МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

І. П. Гамаюн, О. Ю. Чередніченко

МОДЕЛЮВАННЯ СИСТЕМ

Навчальний посібник для студентів спеціальностей
6.050103 “Програмна інженерія”, 6.050101 “Комп’ютерні науки”

Рекомендовано Міністерством освіти і науки України
як навчальний посібник для студенів вищих навчальних закладів

Харків
«Факт»
2015

УДК 004.94
ББК 32.817в6
Г 18

Рекомендовано Міністерством освіти і науки України
як навчальний посібник для студенів вищих навчальних закладів
(лист № 1/11–16122 від 13.10.2014 р.)

Р е ц е н з е н т и:

С. Г. Удовиченко, д-р техн. наук, проф., Харківський національний університет радіоелектроніки;

О. Є. Федорович, д-р техн. наук, проф., Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут».

Гамаюн І. П., Чередніченко О. Ю.

Г 18 Моделювання систем: навчальний посібник для студентів спеціальностей 6.050103 "Програмна інженерія", 6.050101 "Комп'ютерні науки" / І. П. Гамаюн, О. Ю. Чередніченко. – Харків : Факт, 2015. – 228 с.

ISBN 978-966-637-799-2

У посібнику викладені фундаментальні засади моделювання складних систем. Наведена класифікація математичних моделей та методи їхньої побудови. Розглянуто аналітичне моделювання основних видів процесів, що мають місце у складних системах. Описані принципи імітаційного та статистичного моделювання, а також моделювання бізнес-процесів.

Призначено для студентів вищих навчальних закладів, що навчаються за спеціальностями "Програмна інженерія" та "Комп'ютерні науки", викладачів та спеціалістів у галузі комп'ютерного моделювання.

УДК 004.94
ББК 32.817в6

ISBN 978-966-637-799-2

ЗМІСТ

ВСТУП

Швидкий розвиток і ускладнення техніки, збільшення масштабів і вартості робіт при створенні нових систем, широке впровадження автоматизації – все це привело до необхідності введення поняття складної системи (СС).

Складною системою називається множина взаємопов'язаних і взаємодіючих елементів і підсистем різної фізичної природи, що складають нероздільне ціле, забезпечують виконання заданої складної функції і описуються досить складними математичними моделями.

Найбільш важливими у створенні СС є початкові етапи і стадії проектування, де формується уявлення про досліджувану систему, формулюються основні вимоги до тактико-технічних характеристик (ТТХ), створюється узагальнена модель структури системи і описуються закони її функціонування. Помилки, допущені на цих етапах проектування, можуть призвести до невиправданих витрат інтелекту, часу і грошових коштів, нераціональних рішень і, нарешті, навіть до нереалізованих проектів.

Основні особливості СС:

- багатовимірність, пов'язана з наявністю великої кількості елементів і зв'язків між ними;
- ієрархічний багаторівневий характер подання самої системи та структури управління;
- множина структурно-компонувальних рішень;
- багатоаспектність подання;
- розподіленість у просторі та часі;
- багатоцільове функціонування;
- імовірнісне функціонування і поведінка системи, обумовлені складною взаємодією з мінливим зовнішнім середовищем.

Одним з найважливіших інструментів системного аналізу, який дозволяє отримати опис ієрархії зв'язків, механізмів і структур складних багатокомпонентних систем, є моделювання, що дає можливість:

- візуалізувати систему в її поточному або бажаному для дослідника стані;

- визначити структуру або поведінку системи;
- отримати шаблони, що дозволяють формувати і моделювати системи даного класу.

Модельний метод аналізу застосовують для формалізованого опису СС, що містить описи елементарних об’єктів системи, а також їх властивостей і зв’язків. У даний час поряд з широко використовуваними традиційними методами аналітичного моделювання велика роль відведена методам імітаційного моделювання, які дають можливість отримати найбільш якісні результати при аналізі динаміки поведінки СС, дозволяючи найбільш гнучко, повно і наочно відображати процеси, що протікають у них.

До переваг імітаційного моделювання можна віднести:

- динамічний характер відображення системи;
- можливість обліку випадкових факторів і складних залежностей;
- порівняльну простоту введення модифікацій в модель;
- можливість дослідження системи на множині модельних реалізацій її функціонування, тобто проведення статистичних експериментів;
- практично необмежені можливості деталізації моделі залежно від обсягу знань дослідника.

Залежно від характеру модельованих процесів імітаційні моделі (ІМ) поділяють на три великі групи: ІМ безперервних динамічних систем; ІМ дискретних систем; комбіновані, або універсальні ІМ. Крім того, доречна класифікація за принципом динамічного компонування, яка базується на можливості подання об’єкта як сукупності дій або як послідовності подій, або як множини процесів. При цьому найбільш очевидну формалізацію системного підходу до дослідження СС здійснюють на основі поняття "об’єкт як базовий елемент системи". У зв’язку з цим сучасні методи та інформаційні технології моделювання будуються на базі об’єктно-орієнтованого підходу.

В даний час, коли комп’ютерна промисловість пропонує найрізноманітніші засоби моделювання, будь-який кваліфікований інженер або менеджер повинен вміти моделювати СС за допомогою сучасних технологій, реалізованих у вигляді графічних інтегрованих середовищ або пакетів візуального моделювання.

1. ОСНОВНІ ПОЛОЖЕННЯ І ВИЗНАЧЕННЯ ТЕОРІЇ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

1.1. Поняття складної системи та її моделі

Поняття «система» широко використовується в науці і практиці для позначення об'єктів вивчення людини. У науково-технічній літературі, Інтернеті, засобах масової інформації об'єкти різної природи називаються відповідними видами систем – технічної, економічної, політичної, соціальної, операційної і т.д.

Широта уживання поняття системи зумовила і множину визначень цього поняття, в якому виділяють дві групи [1].

Перша група визначень відображає принципи вивчення навколишнього світу, які склалися в природничих науках. Дослідники поділяють досліджувану систему на елементи, вивчають властивості елементів, взаємодію (зв'язки) елементів з урахуванням їх властивостей і таким чином отримують уявлення про систему в цілому як про сукупність взаємопов'язаних елементів.

Одним з представників першої групи є визначення Холла і Фейджи [1]:

«Система – це множина об'єктів разом з відношеннями між об'єктами й між їхніми атрибутами (властивостями)».

Досить точну формалізацію системи *S*, яка визначається згаданим чином, запропонував Месарович [2] шляхом введення сімейства множин $x\widetilde{X} = \{X_i \mid i \in \overline{1,n}\}$, де X_i – повна сукупність прояву атрибуту $i \in \overline{1,n}$. Тоді система *S* є таким відношенням на $\widetilde{X}$, що $S \subset X_1 \times X_2 \times \ldots \times X_n$. Інакше кажучи, система *S* являє собою повну сукупність проявів реально існуючого об'єкта, який називається системою.

Недоліком визначень першої групи є те, що не враховується властивість системи як цілісного об'єкта і властивості, які не зводяться до властивостей елементів системи.

У визначеннях *другої* групи зазначений недолік усувається. У них реалізується принцип подвійності, згідно з яким система, з одного боку, є цілісним об'єктом і як елемент входить у систему більш високого рівня, а з іншого боку, вона розчленовується на елементи, що її складають.

Визначення другої групи «довизначають» систему відповідно до визначень першої групи і тим самим більш повно виражають такі особливості складних систем:

- велика кількість елементів, що утворюють ієрархічну структуру змінного, а іноді і стохастичного характеру;
- дії (функціонування) елементів мають, як правило, невизначений і нестаціонарний характер;
- наявність різних цілей у елементів, які часто мають суперечливий характер;
- відсутність математичного опису, без якого не можна вирішувати, перш за все, завдання управління.

Для того щоб орієнтуватися у величезному різноманітті об'єктів, які визначаються як система, розроблені класифікації систем. Існує багато класифікацій, кожна з яких ґрунтується на певних класифікаційних ознаках: число елементів, характер і тип зв'язків, способи взаємодії з навколишнім середовищем і т.д. Досить всеохоплююча класифікація наводиться в роботах [1, 3, 4]. Відповідно до цієї класифікації системи поділяються на природні, які створені природою, і штучні, які створені людиною. Далі природні поділяються на живі і неживі, а штучні – на формальні і неформальні системи. Важливість цієї класифікації в тому, що в ній чітко виділяється формальна система, або інакше модель системи.

Виникнення формальної системи є результатом взаємодії дослідника з системою як частиною середовища згідно з поставленими цілями. Схема взаємодії показана на рис. 1.1 [1].

Дослідниками називають людей, які виходячи з поставлених цілей вивчають системи, створюють їх опис, пояснюють механізми взаємодії частин і передбачають поведінку системи в часі [1].

Цілі для дослідника формулює людина, що приймає рішення (ЛПР), яка вирішує завдання, пов'язані з управлінням, функціонуванням і розвитком системи. При цьому під розвитком розуміється постановка цілей і планування дій для досягнення цілей, а під функціонуванням – виконання дій, спрямованих на досягнення цілей.

Середовище або суперсистему (метасистему) для системи S утворює сукупність об'єктів, що взаємодіють з S.

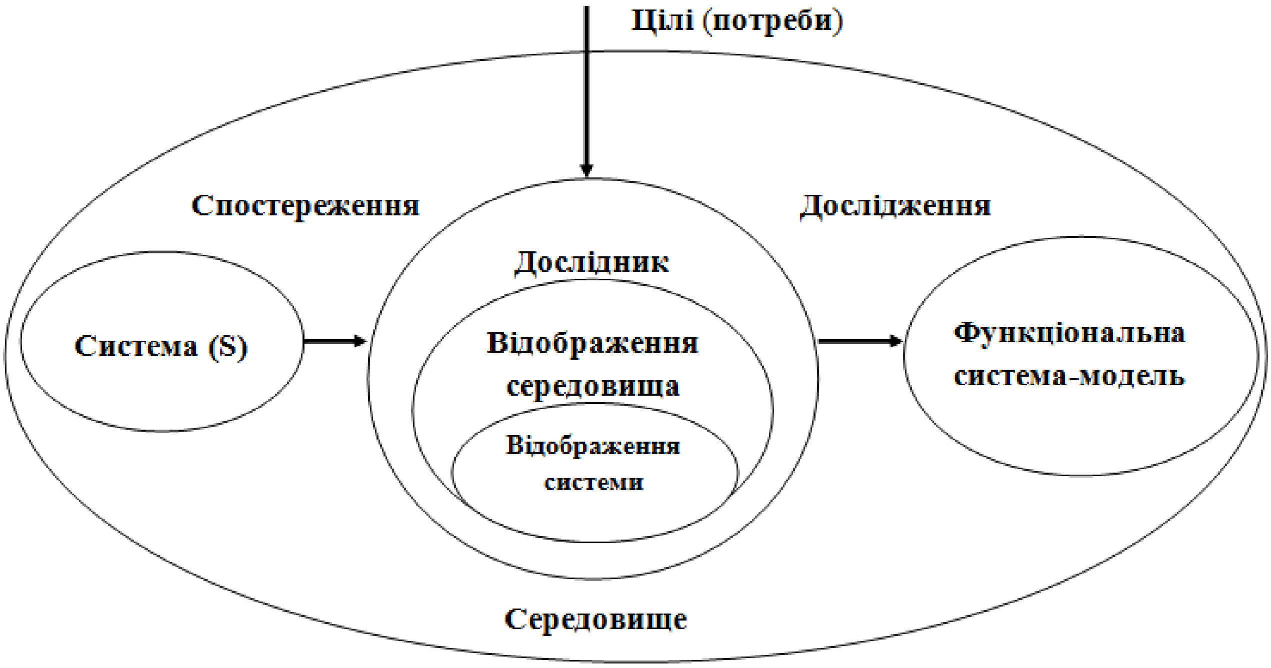


Рисунок 1.1 – Схема взаємодії «система–дослідник»

У взаємодії «система–дослідник» виділяють дії спостереження і дослідження.

Під спостереженням розуміється відображення системи органами почуттів (аналізаторами) дослідника. Спостереження дозволяє створити у дослідника образ системи і середовища, з якими далі виконуються операції аналізу та синтезу. Образи, зафіксовані в пам’яті дослідника, визначаються як представлення.

Дослідження – це процес отримання нових знань про системи, відображених в образах і уявленнях. Дослідження реалізується шляхом виконання етапів опису, пояснення і передбачення.

Опис – це фіксація в знаковій системі уявлень.

Пояснення полягає у формулюванні емпіричних закономірностей і виявленні причинно-наслідкових зв’язків.

Пояснення дозволяє впорядкувати факти в часі, що робить можливим передбачати поведінку системи.

Передбачення – це опис майбутнього системи, її елементного складу, структури та поведінки відносно середовища.

Результатом дослідження, тобто виконання етапів опису, пояснення і передбачення, є створення формальної системи, яка інакше називається моделлю.

Побудова моделі є інтерактивним процесом, реалізація якого спрямована на усунення помилок дослідником на етапах опису, пояснення і прогнозування. Інтерактивний процес передбачає виконання коригування образів і уявлень, що і показано на рис 1.1.

Таким чином, модель є аналогом досліджуваної системи, процесу чи явища, який створюється для визначення властивостей системи, процесу, явища і прогнозування їх поведінки.

Під моделюванням розуміється процес побудови моделі на основі зазначеної на рис. 1.1 схеми взаємодії «дослідник – система».

Теорія моделювання являє собою взаємопов’язану сукупність положень, визначень, методів і засобів створення та вивчення моделей. Ці положення, визначення, методи і засоби, як і самі моделі, є предметом теорії моделювання.

Теорія моделювання є складовою загальної теорії систем – системології, де в якості головного принципу постулюється наступне: система представляється кінцевою множиною моделей, кожна з яких відображає певну сторону (аспект) її сутності.

Модель як формальна система завжди простіше реальної системи, яка є об’єктом вивчення дослідника. У моделі зберігають лише основні якості системи, що вивчається, вони представляють найбільший інтерес для дослідника виходячи з цілей дослідження.

Спрощення моделі в порівнянні з досліджуваною системою дозволяє подолати інформаційний бар’єр складності і провести з моделлю необхідні експерименти. Інформаційний бар’єр складності полягає в тому, що дослідник не у змозі охопити подумки всі множини елементів, зв’язки між ними, події, що відбуваються в системі, яка моделюється.

Очевидно, що моделей не менше, ніж систем, що моделюються. Для того щоб орієнтуватися в множині моделей необхідно їх класифікувати. Одна з повних класифікацій видів моделей наводиться в роботі [6], де в якості класифікаційної ознаки використовується характер процесів у модельованій системі. Схема видів моделей наводиться на рис. 1.2.

На схемі наводяться базові види моделей, з яких можуть складатися інші, більш складні види моделей. Наприклад, такою може бути детермінована, динамічна, дискретна модель.

Детермінована модель не відображає випадкові чинники. *Стохастична* модель, навпаки, призначена для опису процесів з урахуванням випадкових чинників. Статичні моделі відображають стан системи в певний фіксований момент часу, а динамічні – показують зміни стану системи в часі. Очевидно, що динамічна модель подається множиною статичних, кожна з яких відображає стан системи у впорядковані за зростанням моменти часу.

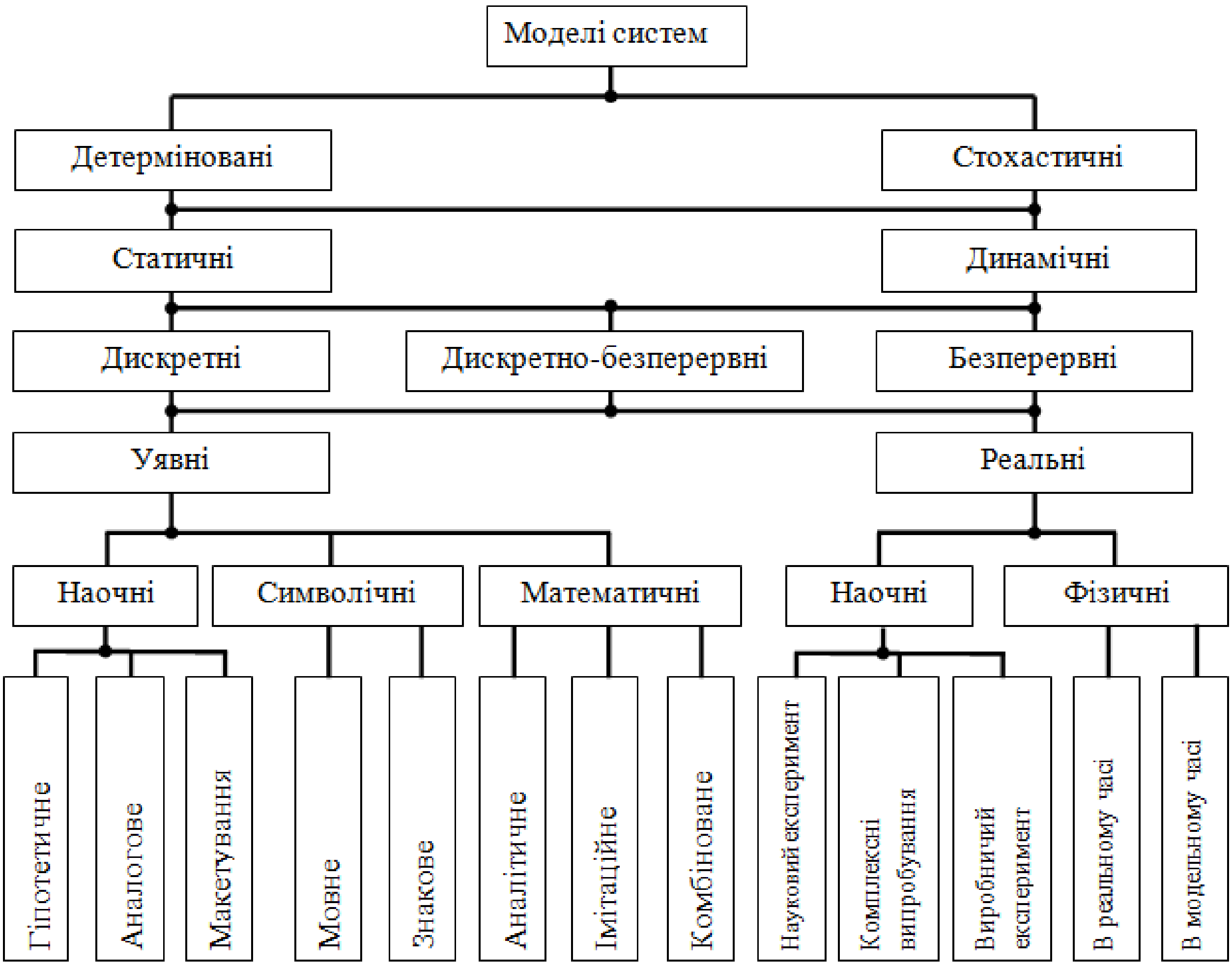


Рисунок 1.2 – Основні види моделей

Дискретні моделі описують процеси, параметри яких або час їх зміни, або те й інше мають дискретний, тобто перериваний характер.

Безперервні моделі описують процеси, параметри яких і час їх зміни безперервні. Приклади зміни значень параметра *x* і часу *t* – *x(t)* дискретного і безперервного процесів показанона рис. 1.3.

Дискретно-безперервні моделі описують системи, в яких одна частина процесів має дискретний, а інша частина – неперервний характер.

У разі дискретного процесу (*а*) значення параметра *x* і часу *t* вибираються з відповідних скінченних множин $\{x_0, x_1, x_2, x_3, x_4, x_5\}$, $\{t_0, t_1, t_2, t_3, t_4, t_5\}$. У разі ж неперервного процесу (*б*) значення параметра *x* і часу *t* належать відповідним множинам, що задаються відрізками $[x_0, x_T]$, $[0, T]$, тобто $x \in [x_0, x_T]$, $t \in [0, T]$.

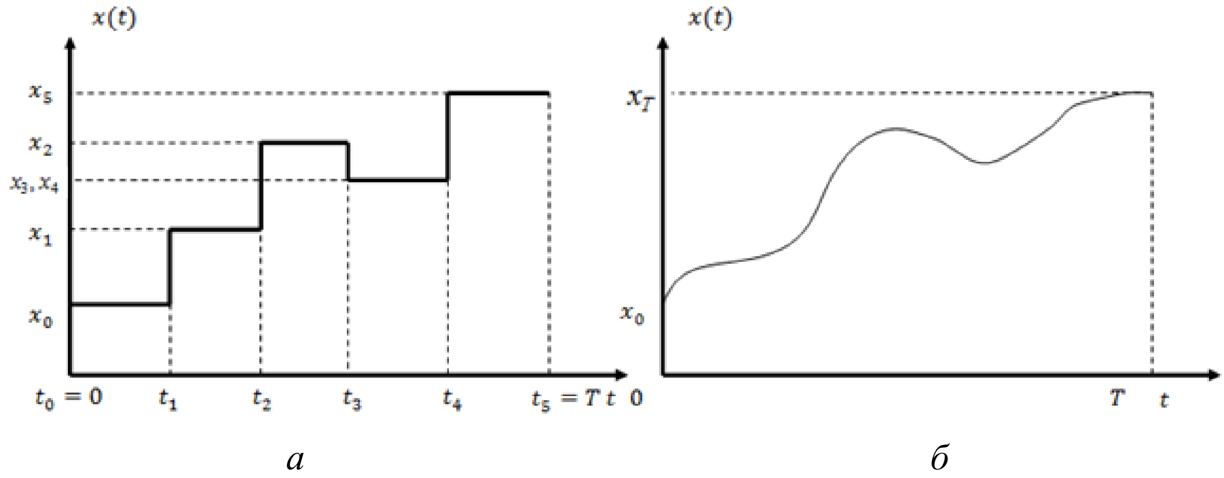


Рисунок 1.3 – Приклади зміни *x(t)* дискретного (*а*) і неперервного (*б*) процесів

Залежно від форми подання модельованої системи виділяються *уявні* (віртуальні) і *реальні* моделі.

Необхідність створення уявних моделей виникає щоразу, коли дослідник відчуває труднощі у фіксації реальних систем в просторі і в часі своїми органами почуттів. Труднощі фіксації в просторі обумовлюються головним чином розмірами досліджуваних систем, які можуть бути об'єктами мікросвіту (атом, молекула і т.д.) або об'єктами макросвіту (планета, Всесвіт і т.д.). Труднощі фіксації в часі зумовлені насамперед випадковим протіканням процесів у досліджуваній системі. Ці процеси можуть тривати протягом часток секунд, а можуть займати сторіччя.

Основними видами уявного моделювання є натурне, символічне і математичне.

Натурна модель створюється на основі уявлень дослідника про реальну систему у вигляді деякого наочного образу – малюнка, графіка, макета і т.д. Натурне моделювання підрозділяється на гіпотетичне, аналогове і макетування.

В основу *гіпотетичного* моделювання закладається деяка гіпотеза про закономірності протікання процесів у реальній системі. При цьому під гіпотезою розуміється передбачення, яке ґрунтується на невеликому числі дослідних даних, спостережень, здогадів.

Аналогове моделювання ґрунтується на аналогіях різних рівнів. Вищим рівнем є повна аналогія, яка може застосовуватися при моделюванні простих об'єктів. З ускладненням об'єктів використовуються аналогії нижчих рівнів. У цьому випадку аналогова модель відображає кілька або навіть один аспект модельованої складної системи.

Уявне *макетування* застосовується в тих випадках, коли процеси в реальній системі не відтворюються на установках, які зберігають природу явищ і володіють фізичною схожістю. Уявні чи уявлені макети ґрунтуються на аналогіях і причинно-наслідкових зв'язках між явищами і процесами в модельованій системі.

Символічні моделі виражають основні властивості модельованої системи за допомогою знаків або символів.

У *знаковій* моделі для позначення понять використовуються знаки, які зв'язуються операціями алгебри логіки. За допомогою знаків і операцій будуються слова, пропозиції, висловлювання, які описують модельовану систему.

У *мовному* моделюванні використовується тезаурус, який від звичайного словника відрізняється відсутністю невизначеностей, тобто кожне поняття визначається одним словом.

Основним засобом для дослідження складних систем є *математичні* моделі, під якими розуміються формальні системи з елементами у вигляді математичних об'єктів (чисел, змінних, матриць, множин і т.д.) і відносин між ними, що відображають властивості модельованої реальної системи.

Вид математичної моделі залежить від природи модельованої системи, цілей моделювання, необхідної точності та достовірності вирішення завдань дослідження і т.д. У класифікації на рис 1.2 математичні моделі поділяються на аналітичні, імітаційні і комбіновані.

Аналітичне моделювання можливе, коли функціонування елементів модельованої системи вдається подати функціональним співвідношенням деякого вигляду: алгебраїчним, інтегральним, диференціальним, скінченно-різницевим.

Аналітична модель дозволяє провести дослідження модельованої системи трьома основними методами: аналітичним, чисельним, якісним. Аналітичний метод полягає в отриманні в явному вигляді залежності між параметрами досліджуваної системи.

Наприклад, якщо аналітична модель являє собою диференціальне рівняння

$$\frac{d^2y}{dt^2} + \omega^2 y(t) = 0,$$

то реалізація аналітичного методу дослідження зводиться до розв'язання даного диференціального рівняння:

$$y(t) = y_0 \cos \omega t + \frac{\dot{y}_0}{\omega} \sin \omega t,$$

де $y_0 = y(0)$; $\dot{y}_0 = \dot{y}(0)$ – початкові умови розв'язання диференціального рівняння.

Чисельний метод використовується, коли реалізація аналітичного методу неможлива або супроводжується значними труднощами і полягає в отриманні деякого часткового розв’язання для конкретних початкових даних.

Якісний метод використовується у випадку, коли неможливо реалізувати аналітичний або чисельний метод. Він не спрямований на отримання саме розв’язання, а забезпечує можливість визначення деяких його властивостей. Наприклад, за допомогою граничного переходу можна іноді встановити значення, до якого прагне параметр досліджуваної системи.

Імітаційна модель – це алгоритм або сукупність алгоритмів, що відображають процес функціонування модельованої системи в часі.

Основною перевагою імітаційного моделювання в порівнянні з аналітичним є можливість адекватного відображення більш складних систем. Ця можливість забезпечується з урахуванням таких характерних особливостей складних систем як: дискретність, лінійність, невизначеність, нестаціонарність та інші, що складно врахувати в аналітичних моделях.

Комбіноване моделювання передбачає використання аналітичного та імітаційного моделювання. У складній системі виокремлюють частину процесів, яку описують за допомогою аналітичних моделей, і іншу частину процесів, яку описують за допомогою імітаційних моделей. При цьому головним є найбільш повне використання переваг кожного з цих видів моделювання.

Реальне моделювання передбачає можливість дослідження різних характеристик за допомогою або самої реальної системи, або її частини. Дослідження проводяться як в нормальному режимі функціонування, так і в спеціальному режимі, який передбачає, наприклад, створення екстремальних умов (висока або низька температура, високий тиск або вакуум і т.д.). Реальне моделювання є найбільш адекватним, але його використання для дослідження складних систем дуже обмежене, а для систем, що проектуються, є неможливим.

Реальне моделювання відповідно до схеми на рис 1.2 розділяється на наочне і фізичне.

Натурним моделюванням передбачається проведення досліджень на реальному об’єкті або його копії з подальшим відпрацюванням результатів експериментів на основі теорії подібності. Добре відомим прикладом натурного моделювання є створення зменшеної копії літака або автомобіля, яку поміщають в аеродинамічну трубу, де досліджуються аеродинамічні властивості копії. Отримані результати перетворюються на основі теорії подібності і таким чином отримують аеродинамічні характеристики реального об’єкта.

З визначення натурного моделювання виходить важлива роль експерименту або з самим реальним об’єктом, або з його копією. Тому в наочному моделюванні виділяються види, які відповідають різним видам експерименту. Відповідно до класифікації на рис. 1.2 виділяють такі види: науковий експеримент, комплексні випробування, виробничий експеримент.

Основним завданням наукового експерименту є пізнання сутності процесів за допомогою засобів автоматизації проведення експерименту та обробки інформації. Експеримент відрізняється від звичайного, нормального ходу процесу тим, що вводяться нові фактори, які дозволяють досліджувати поведінку реального об'єкта в екстремальних умовах і визначати межі стійкості.

Різновидом експерименту є комплексні випробування, в ході яких вдається зібрати досить великий обсяг статистичних даних і обробити їх, що дозволяє виявити закономірності таких якісних характеристик як точність, надійність, швидкодію і т.д.

Виробничий експеримент полягає в узагальненні досвіду впровадження та експлуатації зразків нової техніки або технології. Формою проведення виробничого експерименту є наради, семінари та ін.

Останній вид моделювання, зазначений на рис. 1.2, – це фізичне моделювання. Цей вид моделювання відрізняється від наочного тим, що експерименти проводяться не на реальних об'єктах, а на установках, які зберігають і відтворюють природу явищ, що цікавлять дослідника. Фізичне моделювання може проводитися в реальному часі і в спеціальному масштабі часу, який називають модельним часом.

1.2. Вимоги до моделей складних систем

Простота моделі в порівнянні з реальною модельованою системою, як зазначалося, дозволяє подолати інформаційний бар'єр складності і провести з моделлю необхідні експерименти. Метою планованих експериментів є визначення властивостей моделі і на їх основі властивостей системи.

Спрощення моделі в порівнянні з модельованої системою є однією з основних вимог моделювання. Однак ця вимога повинна виконуватися разом з іншими основними вимогами універсальності, адекватності, точності та економічності, що висуваються до моделей.

Під ступенем універсальності моделі розуміється повнота відображення тих якостей досліджуваної системи, які визначаються метою дослідження.

Для визначення інших вимог використовуються зовнішні, внутрішні та вихідні параметри. Вектор зовнішніх параметрів Z відображає властивості зовнішнього середовища. Вектор внутрішніх параметрів *X* відображає властивості елементів, з яких система складається, або інакше – відображає внутрішній зміст системи. І нарешті, вектор вихідних параметрів *Y* відображає властивості системи відносно інших систем і зовнішнього середовища.

Між *X*, *Y*, *Z* має місце функціональне співвідношення:

$$Y = F(X, Z), \tag{1.1}$$

яке є математичною моделлю системи при відповідному визначенні F.

Нехай Y_s – вектор вихідних параметрів на виході системи, а Y_m – вектор вихідних параметрів на виході моделі системи.

Тоді відносна похибка визначається як

$$\varepsilon = \frac{|Y_m - Y_s|}{Y_s}. \tag{1.2}$$

Для зведення отриманої векторної оцінки відносної похибки до скалярної величини використовують деяку норму вектора, прикладами якої можуть бути:

$$\|\varepsilon\| = \sqrt{\sum_{j \in \overline{1,n}} \varepsilon_j^2}\ ;\ \|\varepsilon\| = \max_{j \in \overline{1,n}} \varepsilon_j\,,$$

де ε_j $(j \in \overline{1,n})$ – компоненти вектора ε.

Під адекватністю моделі розуміється здатність відображати задані властивості досліджуваної системи із заданою похибкою, граничне значення якої може задаватися константою $\delta > 0$.

З виразів (1.1), (1.2) випливає, що значення норми $\|\varepsilon\|$ залежить від вектора *X* і *Z*. Важливим інженерним завданням є визначення *X*, мінімізуючого $\|\varepsilon\|$ при заданому векторі *Z*. Якщо позначити $\|\varepsilon\| = f(X, Y)$, то зазначена задача формалізується записом

$$\min_{\{x\}} f(X, \overline{Z}) = \varepsilon_m\,,$$

де $\overline{Z}$ – заданий вектор *Z*, тобто конкретний набір значень параметрів зовнішнього середовища.

Очевидно, що при зміні вектора $\overline{Z}$ отримуються різні значення ε_m. Множина $\overline{Z}$ і відповідна йому множина значень ε_m використовуються для побудови області адекватності, що визначається як

$$\{\overline{Z} \mid \varepsilon_m < \delta\}.$$

Економічність моделі характеризується витратами ресурсів комп’ютера на її реалізацію. При цьому до ресурсів насамперед належать машинний час і оперативна пам’ять. Непрямими показниками витрат цих ресурсів є число операцій, необхідних для звернення до моделі, розмірність і число внутрішніх параметрів моделі.

Вимоги точності, універсальності, адекватності, як правило, суперечать вимозі економічності. Це протиріччя вирішується шляхом компромісного рішення, яке досягається реалізацією узгоджених, взаємних поступок у вимогах.

1.3. Класифікація математичних моделей

За класифікаційною схемою (рис. 1.2) математичні моделі поділяються на аналітичні, імітаційні і комбіновані. Це є поділ за класифікаційною ознакою «спосіб подання властивостей модельованої системи».

Існує і багато інших класифікаційних схем, кожна з яких створюється на основі певної класифікаційної ознаки. Нижче розглядаються деякі з них.

Так, за класифікаційною ознакою «характер відображуваних властивостей модельованої системи» математичні моделі поділяються на структурні і функціональні.

Структурні моделі відображають відповідні структурні властивості модельованої системи і поділяються на топологічні і геометричні.

За допомогою топологічних моделей відображається склад і взаємні зв'язки елементів модельованої системи. Цей вид моделей часто використовується для розв'язання завдань, в яких суттєвим є просторова прив'язка елементів до певних позицій і/або тимчасова прив'язка елементів до певних моментів часу. Прикладами таких завдань можуть бути завдання компонування обладнання на виробничих площах, розміщення деталей у виробах машинобудування чи приладобудування, трасування з'єднань у різних схемах, складання розкладів виконання технологічних процесів, розподілу повноважень щодо прийняття рішень між особами або групами осіб (структурними підрозділами) і багато інших завдань . Всі ці вони є завданнями структурного синтезу. Їх розв'язанням є варіант структури системи, під якою розуміється організація системи з окремих елементів з їх взаємозв'язками, що визначаються розподілом функцій і цілей, виконуваних системою [7].

Досить загальною є наступна формальна постановка задачі структурного синтезу [7]:

$P=\{\pi\}$ – множина наборів можливих принципів побудови системи або її елементів;

$F(\pi) = \{f\}$ – множина взаємопов’язаних функцій (операцій), виконуваних системою, яка залежить від набору принципів побудови системи;

$ST = \{str\}$ – множина варіантів структур як множина можливих взаємопов’язаних елементів системи, якими можуть бути технічні засоби, пункти обслуговування, окремі виконавці або структурні підрозділи організації і т.д.

Нехай Ф – відображення множини $F(\pi)$ в множині ST, тоді Ф: $F(\pi) \rightarrow ST$. Шуканим варіантом може бути $str = \varPhi(f)$. Варіант структури є оптимальним, якщо досягається екстремальне значення деякої цільової функції при заданих обмеженнях.

Відомі кілька рівнів складності задач структурного синтезу [8]. Більшість з них відносять до третього рівня складності і класу так званих NP-повних задач. Це клас комбінаторних задач, для точного розв’язання яких відомі алгоритми тільки експоненціальної складності, тобто алгоритми, в яких кількість необхідних операцій пропорційна експоненті від розмірності задачі.

Топологічні моделі часто мають форму графів, таблиць, матриць, списків і т.п.

Інший вид структурних моделей – геометричні моделі – використовуються для відображення геометричних властивостей об’єктів моделювання. У них додатково до відомостей про взаємне розташування елементів наводиться інформація про форму елементів, що складають систему.

Геометричні моделі продаються рівняннями, які описують лінії і поверхні елементів; алгебрологічними співвідношеннями, які описують області, що становлять тіло елемента; графами і списками, що відображають конструкції з типових конструктивних елементів [9].

Теоретичною основою створення геометричних моделей є аналітична геометрія, теорія множин, диференціальна геометрія, теорія графів, алгебра логіки.

Геометричні моделі мають свою класифікацію [10]. Виділимо в ній найбільш відомі і широко використовувані види: аналітичні та алгебраїчні геометричні моделі.

Прикладом аналітичної геометричної моделі може служити загальне рівняння кривої другого порядку на площині в прямокутній системі координат

$$F(x, y) = ax^2 + 2bxy + cy^2 + 2dx + 2ey + g = 0,$$

де a, b, c, d, e, g – постійні коефіцієнти.

Крива $F(x, y)$ використовується для опису контуру елемента. Для опису поверхні обертання може бути використане рівняння

$$F(x, y, z) = a(x^2 + y^2) + bz^2 + 2cz + d = 0,$$

де *x*, *y*, *z* – осі координат.

Аналітичні моделі, як правило, служать для опису елементарних геометричних об’єктів, на основі яких можуть бути отримані складні геометричні об’єкти.

За допомогою алгебрологічних геометричних моделей описуються плоскі фігури і тіла в тривимірному просторі. Модель являє собою логічну функцію умов належності точок заданої області.

Нехай, наприклад, необхідно розробити модель, що описує плоску фігуру у вигляді трикутника *D* на рис. 1.4.

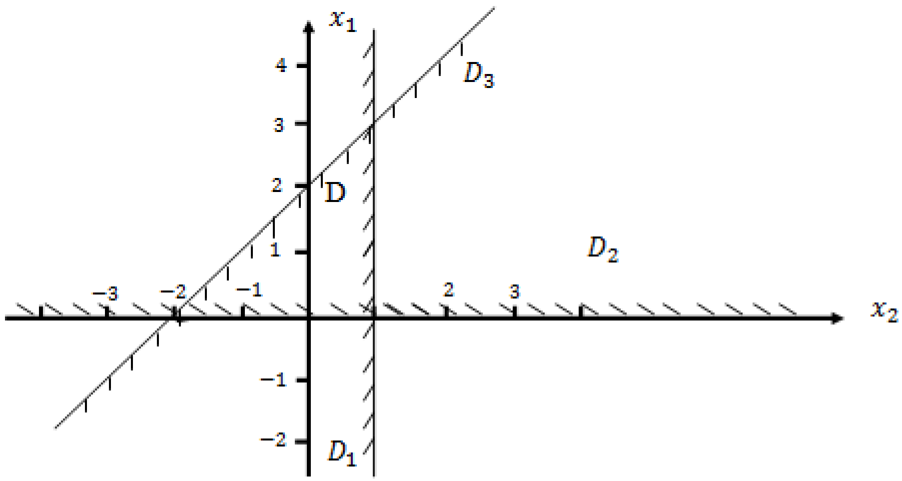


Рисунок 1.4 – Плоска фігура *D* у просторі $x_1 0 x_2$

Моделлю фігури *D* є логічний вираз

$$D = D_1 \cap D_2 \cap D_3,$$

де області D_1, D_2 і D_3 визначаються нерівностями:

$$D_1 : x_2 \leq 1,$$

$$D_2 : x_1 \geq 0,$$

$$D_3 : x_1 - x_2 \leq 2.$$

Інші види геометричних моделей – канонічні, каркасні, рецепторні, математичні і т.д. – розглядаються детально в спеціальній літературі [10].

На відміну від структурних, функціональні моделі відображають фізичні або інформаційні процеси в модельованій системі при її функціонуванні або виготовленні.

Оскільки функціональні моделі відображають протікання процесів у системі, що функціонує, то для класифікації цих моделей повною мірою може бути використана класифікація за характером процесів, яка наводиться на рис. 1.2. Тобто, як і процеси, функціональні моделі поділяються на дискретні, неперервні, стохастичні і т.д.

Визначення виду моделі, виходячи з характеру процесу, дозволяє вибрати математичний апарат або схему для побудови моделі. Наприклад, для неперервно-детермінованих процесів моделі створюються на основі математичного апарату диференціальних рівнянь, для дискретно-детермінованих процесів – часто на основі апарату скінченних автоматів і т.д.

Крім характеру процесу, для визначення функціональної математичної моделі важливим є фізична природа процесу, або інакше аспект опису. Виділення аспектів опису призводить до виділення моделей електричних, механічних, гідравлічних, оптичних, хімічних і т.п. Зазначені аспекти дозволяють вибрати закони та закономірності перебігу процесів. Ці закони складають основу для побудови функціональної моделі. Наприклад, часто основою для моделей механічних систем є закони Ньютона, Гука і т.д., для електричних систем – закони Ома, Кірхгофа і т.д.

Нехай, наприклад, необхідно розробити функціональну модель такої системи як пружинний маятник на рис. 1.5.

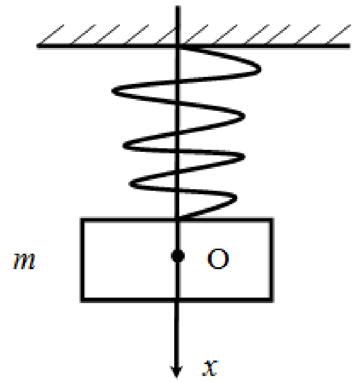


Рисунок 1.5 – Вигляд пружинного маятника в положенні рівноваги

Система складається з пов’язаних між собою двох елементів: пружини з коефіцієнтом жорсткості k і маси m. Рух маси в результаті стиснення і розтягування пружини здійснюється вздовж осі x, на якій через «0» позначається положення рівноваги. Силами опору нехтуємо.

Для розглянутої системи найбільш важливим є процес її руху, який має неперервно-детермінований характер. Отже, для побудови математичної функціональної моделі руху даної системи може бути використаний математичний апарат або схема звичайних диференціальних рівнянь.

Щоб більш конкретно визначити математичну модель, необхідно встановити фізичну природу (аспект) процесу руху і закони цього руху. За своєю природою розглянута система механічна. Законом, що визначає динаміку системи, є 2 закон Ньютона, згідно з яким

$$a \cdot m = F, \tag{1.3}$$

де a – прискорення, F – сила розтягування і стиснення пружини, що діє на масу m.

Відповідно до закону Гука $F = -kx$, де x – відхилення маси m від положення рівноваги.

Раніше йшлося, що для побудови моделі необхідно використовувати апарат (схему) диференціальних рівнянь. Тому прискорення a подається як $a = \dfrac{d^2x}{dt^2}$. З урахуванням цього, вираз (1.3) набуває вигляду

$$\frac{d^2x}{dt^2}m = -kx.$$

Увівши відоме позначення $\omega^2 = \dfrac{k}{m}$ (ω – частота коливань), отримуємо остаточний вид функціональної математичної моделі руху (динаміки) розглянутої системи у вигляді пружинного маятника:

$$\frac{d^2x}{dt^2} + \omega^2 x(t) = 0.$$

Таким чином, моделлю є звичайне, однорідне, диференціальне рівняння другого порядку.

Ієрархічність структури існуючих складних систем і блочно-ієрархічний принцип їх проектування зумовлюють ієрархію відповідних математичних моделей.

Залежно від місця в ієрархії описів функціональні та структурні математичні моделі поділяються зазвичай на три види, кожен з яких відповідає одному з трьох узагальнених рівнів: мікро-, макро-, метарівням. При цьому модель більш високого рівня є, як правило, агрегованою.

Формально агрегування полягає в заміні деякої групи змінних, що відображають стан системи, однією змінною. Фізично агрегування відповідає укрупненій дискретизації простору за функціональною ознакою.

На метарівні в якості елементів можуть бути досить великі сукупності елементів нижчих рівнів.

У достатньо загальному вигляді задача побудови агрегованої моделі може бути сформульована таким чином [8].

Нехай вихідна багатовимірна модель являє собою систему рівнянь

$$y_i = f_i(x_1, x_2, \ldots, x_n),\ i \in \overline{1,n}, \tag{1.4}$$

де $x = (x_1, x_2, \ldots, x_n)$ – входи, а $y = (y_1, y_2, \ldots, y_n)$ – виходи моделі.

Агрегування полягає у введенні агрегованих змінних і системи рівнянь, що пов’язують їх.

Агреговані змінні виражаються через змінні вихідної моделі:

$$z_k = \psi_k(x_1, x_2, \ldots, x_n);$$

$$u_k = \varphi_k(y_1, y_2, \ldots, y_n),\ (k \in \overline{1,m}),\ (m < n). \tag{1.5}$$

Система рівнянь, що зв’язує агреговані змінні, має вигляд:

$$u_k = s_k(z_1, z_2, \ldots, z_k),\ k \in \overline{1,m}, \tag{1.6}$$

де z_k, u_k слід розглядати відповідно як входи і виходи агрегованої моделі.

Якщо позначити $x = (x_1, x_2, \ldots, x_n)$, то умовою адекватного агрегування є

$$\forall x \in X \quad \varphi_k\big(f_1(x), f_2(x), \ldots, f_n(x)\big) = s_k\big(\psi_1(x), \psi_2(x), \ldots, \psi_m(x)\big), \tag{1.7}$$

де X – множина входів вихідної моделі (1.4).

Умова адекватного агрегування по суті є умовою спільності системи агрегування (1.5), (1.6) для вихідної моделі (1.4), а система агрегування, що задовольняє цій умові, називається спільною системою агрегування.

Побудова спільної системи агрегування вимагає визначення додаткових умов, що накладаються на функції f, ψ, φ, s. Ці додаткові умови призводять до великих обчислювальних витрат і часто не мають фізичної інтерпретації.

Відмова від задоволення умов спільності породжує велике число критеріїв і способів агрегування, серед яких виділяються методи агрегування лінійних моделей, методи дескриптивного агрегування, методи мінімізації функції від помилки агрегування і від помилки спрощення [8].

Методи агрегування лінійних моделей базуються на аналізі матриць коефіцієнтів вихідної моделі (1.4), що дозволяє виділити межі змінних, кожною з яких може бути одна агрегована змінна.

При дескриптивному агрегуванні фіксуються вихідна модель f і функції агрегування ψ, φ (1.5), а агрегована модель s (1.6) повинна задовольняти умові (1.7) не для всіх $x \in X$, а лише для деякого базового вектора $x^0 \in X$.

При реалізації методів мінімізації помилки агрегування під помилкою зазвичай розуміється різниця між правою і лівою частинами умови сумісності (1.7).

Використання спрощеної моделі як основи методу мінімізації функції від помилки спрощення доцільно у випадках, коли вихідна модель (1.4) досить складна з обчислювальної точки зору. Помилка спрощення визначається як різниця між виходами спрощеною і вихідної моделей.

1.4. Методи і методика процесу побудови математичної моделі

Для побудови математичної моделі використовуються неформальні і формальні методи [8].

Неформальні методи використовуються для отримання математичних моделей елементів системи, що розташовуються на різних ієрархічних рівнях.

Реалізація неформальних методів включає вивчення закономірності процесів і явищ у модельованій системі або її елементах, виділення істотних факторів, прийняття різного роду припущень та їх обґрунтування, математичну інтерпретацію наявних відомостей і т.д.

Усі зазначені операції побудови математичної моделі реалізуються кваліфікованими фахівцями. Від успішності реалізації цих операцій залежать показники ефективності математичної моделі – ступінь універсальності, точність, економічність.

Метою вивчення закономірностей процесів і явищ є визначення природи процесів, явищ і законів, яким вони підкоряються. Приклад реалізації цієї операції вже розглядався при побудові моделі пружинного маятника, вигляд якого показаний на рис. 1.5.

Виділення істотних факторів є важливою неформальною операцією, реалізація якої можлива за допомогою фахівців, які добре обізнані у предметній області. З практики побудови математичних моделей відомо, що 20 % факторів визначають 80 % властивостей модельованої системи, а інші 80 % чинників визначають 20 % властивостей. Проблема полягає у визначенні цих 20 % істотних факторів. Їх визначення реалізується за допомогою експертних методів, якими передбачається отримання експертних оцінок і їх обробка методами статистики з введенням нових коефіцієнтів, що враховують важливість оцінки того чи іншого експерта [12].

Прийняття різних припущень та їх обґрунтування використовується головним чином для спрощення моделі. Часто використовується спрощення, що полягає в заміні нелінійної моделі реального процесу деякою сукупністю лінійних моделей. Суть цієї заміни показано на рис. 1.6.

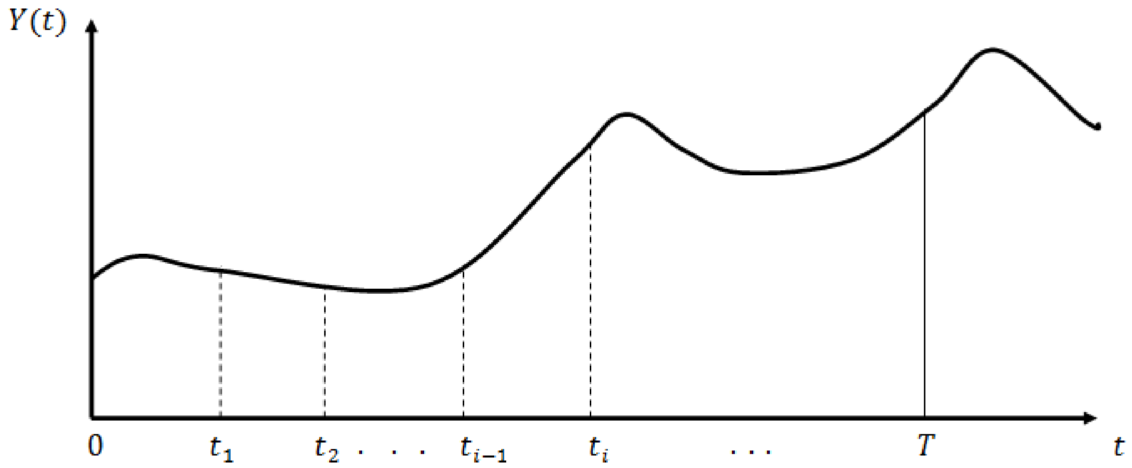


Рисунок 1.6 – Приклад зміни характеристики $y(t)$

Вихідна характеристика $y(t)$ на інтервалі $[0,T]$ має явно виражений нелінійних характер. При цьому закономірність $y(t)$ досить складна і тому виникає необхідність у її спрощенні. Для цього інтервал розбивають на менші інтервали $[t_{i-1},t_i]$ $\left(i \in \overline{1,n},\, t_0 = 0, t_n = T\right)$, і робиться припущення про лінійність закону зміни y на інтервалі, тобто $\tilde{y}_i(t) = a_i + b_i t$, $t \in [t_{i-1},t_i]$. Обґрунтуванням такого припущення є виконання умови

$$\forall t \in [t_{i-1},t_i] \quad |y(t) - \tilde{y}_i(t)| < \varepsilon, \tag{1.8}$$

де ε – допустима похибка відхилення нелінійної характеристики $y(t)$ від лінійної $\tilde{y}_i(t)$ на інтервалі $[t_{i-1},t_i]$.

Якщо умова (1.8) виконується для всіх інтервалів $i \in \overline{1,n}$, то одна нелінійна характеристика реального процесу може замінюватися сукупністю з n лінійних моделей $\{\tilde{y}_i(t) = a_i + b_i t \mid i \in \overline{1,n}\}$.

Під математичною інтерпретацією наявних відомостей розуміється емпірична інтерпретація як тлумачення об’єктів формальних і змістовних теорій у категоріях реального світу [12, 13]. У процесі емпіричної інтерпретації, яка характерна для прикладної математики, зв’язуються поняття теорії з об’єктами і відношеннями фізичного простору. Наприклад, однією з широко використовуваних теорій прикладної математики є теорія графів, основними поняттями якої є вершина, ребро, дуга. Пов’язуючи ці поняття з населеними пунктами і дорогами або з елементами електричних схем (резистор, конденсатор тощо) і з'єднаннями між ними, створюються графові моделі для дослідження відповідно транспортних систем або електричних ланцюгів.

Неформальні методи використовуються для розробки як теоретичних, так і емпіричних математичних моделей. Теоретичні моделі є результатом дослідження процесів та їх закономірностей, які характерні для розглянутого класу об'єктів і явищ.

Емпіричні моделі створюються для вивчення властивостей системи, які вона проявляє відносно зовнішнього середовища, тобто відносно оточуючих її систем. Ці моделі часто належать до класу «чорний ящик» вигляду

$$Y = F(Z), \tag{1.9}$$

основна відмінність яких від моделей (1.1) полягає в тому, що не враховується внутрішній устрій системи, який виражається вектором внутрішніх параметрів системи *X*.

Для визначення *F* у моделі (1.9) проводяться експерименти, які полягають у зміні за заздалегідь розробленим планом вектора *Z* і фіксації відповідних значень вектора *Y*. У результаті отримують так звані «статистики моделювання» $\{Y_j, Z_j \mid j \in \overline{1,m}\}$, де *j* – індекс експерименту, *m* – загальне число експериментів, Y_j, Z_j – вектори *Y*, *Z* у *j*-му експерименті.

Обробка статистики моделювання дозволяє визначити *F*. При обробленні спочатку висувається гіпотеза про структуру моделі (про вигляд моделі), а потім визначаються параметри моделі часто на основі методу найменших квадратів [11].

Таким чином, у цілому методикою одержання математичних моделей елементів системи на основі неформальних методів передбачається виконання послідовності наступних дій.

1. Визначаються властивості, які необхідно відобразити в моделі. Вибір властивостей залежить від цілей і задач моделювання і визначає ступінь універсальності математичної моделі.

2. Підготовка інформації щодо кожної властивості, обраної в п.1. Джерелами інформації є бази даних і знань, науково-технічна література з описом раніше виконаних розробок подібних математичних моделей, результати експериментів, досвід і знання експертів і т.д.

3. Розробка структури математичної моделі, під якою розуміється загальний вигляд математичних співвідношень без вказівки чисельних значень параметрів моделі. Структура St і вектор параметрів C є парою моделі F в (1.1), тобто

$$F = \langle St, C \rangle. \qquad (1.10)$$

Відомо, що під структурою розуміється деякий набір елементів, з яких складається об’єкт, що вивчається, з відносинами між елементами.

Структури багатьох реальних об’єктів подаються графом, вершинам якого відповідають елементи об’єкта, а дугам – відношення між елементами.

Для задання структури математичної моделі в якості елементів використовуються такі категорії як розмірність, лінійність, нелінійність, детермінованість, стохастичність, дискретність, неперервність і т.д. Структура моделі задається перерахуванням зазначених категорій. Наприклад, двовимірна, статична, детермінована, лінійна, неперервна, стаціонарна модель (1.1) має вигляд

$$y = C_1 + C_2 x + C_3 z, \qquad (1.11)$$

де x, y, z – скалярні величини, C_1, C_2, C_3 – постійні коефіцієнти, значення яких відомі.

Якщо категорію статична замінити на категорію динамічна, то (1.11) може отримати вигляд

$$y(t) = C_1 + C_2 x(t) + C_3 z(t).$$

Якщо до того ж змінити ще категорію стаціонарна на нестаціонарна, то (1.11) приймає вигляд

$$y(t) = C_1(t) + C_2(t) x(t) + C_3(t) z(t),$$

тобто і параметри стану, і параметри, що визначають властивості системи або елементи, є функціями часу.

4. Розрахунок числових значень параметрів C математичної моделі в парі (1.10). Задачу визначення параметрів $C = (c_1, c_2, ..., c_k)$ у режимі нормального функціонування модельованого об’єкта, тобто без подачі на об’єкт спеціальних керуючих впливів, називають задачею ідентифікації параметрів моделі.

Для розв’язання завдання ідентифікації використовується інформація про структуру St та спостереження за входом і виходом об’єкта моделювання при його взаємодії із зовнішнім середовищем. Якщо зафіксувати властивості елементів системи, що виражаються вектором X, то для визначення параметрів C моделі (1.1) необхідна інформація падається парою

$$I = \langle Z_i, Y_i \rangle, \ (i \in \overline{1, n}),$$

де Z_i, Y_i – вектори в момент спостереження $t_i \in [0, T]$ з інтервалу $[0, T]$, n – число спостережень.

Визначення C зводиться до реалізації алгоритму на вихідних даних у вигляді структури St і спостережень I, тобто

$$C = \varphi(St, I). \qquad (1.12)$$

Алгоритми φ розв'язання задачі ідентифікації поділяються на два великі класи – адаптивні і неадаптивні.

Під адаптивним алгоритмом φ_a розуміється алгоритм, що дозволяє отримати значення параметрів C_{i+1} на $(i+1)$ кроці алгоритму за значеннями параметрів C_i на попередньому i-му кроці і інформації про значення входів і виходів об'єкта моделювання на $(i+1)$ кроці $I_{i+1} = \langle Y_{i+1}, Z_{i+1} \rangle$, тобто

$$C_{i+1} = \varphi_a(C_i, I_{i+1}).$$

Таким чином, адаптивний алгоритм дозволяє отримати значення параметрів шляхом реалізації послідовності з n кроків. При цьому на кожному кроці використовується тільки інформація про стан входів і виходів на даному кроці і значення параметрів на попередньому кроці алгоритму.

На відміну від адаптивного, неадаптивний алгоритм припускає використання відразу всієї інформації про стан входів, виходів об'єкта моделювання $I = \langle Z_i, Y_i \rangle \; i \in \overline{1, n}$, тобто неадаптивний алгоритм є по суті алгоритмом φ в (1.12). Прикладом неадаптивного алгоритму є алгоритм на основі методу найменших квадратів [13].

Оцінення трудності та адекватності математичної моделі на основі (1.2) з побудовою області адекватності та їх різних апроксимацій розглянуто в [12].

Формальні методи використовуються головним чином для побудови математичних моделей систем на основі наявних моделей елементів системи та зв'язків між елементами. При цьому часто моделями елементів є компонентні рівняння, а зв'язки елементів один з одним враховуються за допомогою топологічних рівнянь [8].

1.5. Ядро знань з моделювання

Ядро знань – body of knowledge (BOK) – з моделювання та імітаційного моделювання – modeling and simulation (M&S) – створюється під керівництвом Національної асоціації навчальних систем (National Training Systems Association – NTSA). Ядро знань M&SBOK відкрите для професіоналів в області моделювання. З метою редагування та удосконалення M&SBOK створений спеціальний наглядовий комітет. Члени цього комітету відомі і відкриті для спілкування та обміну думками.

Для того щоб підтримувати такий інтерактивний підхід до створення M&SBOK, був організований механізм отримання рекомендацій від колег у галузі моделювання та імітаційного моделювання.

Можна виділити кілька аспектів розуміння того, що таке ядро знань M&SBOK. З прагматичної точки зору, ядро знань – це:

1) структуровані знання, що використовуються представниками даної дисципліни в якості рекомендацій в практиці і роботі;
2) задана сукупність знань у певній галузі, які повинен опанувати спеціаліст для того, щоб кваліфікуватися як практик;
3) стартовий майданчик для об'єднання деякого співтовариства.

В деякій мірі розвиток M&SBOK обґрунтовується існуванням ядер знань з інших дисциплін. Деякі дисципліни мають кілька ядер знань для подання своїх різних напрямків. Наприклад, у рамках управління якістю можна виділити наступні:

- ядро знань з якості;
- ядро знань з якості та підвищення продуктивності;
- ядро знань для Екзаменаційної комісії Національної ради інспекторів;
- ядро знань по Шість Сигм (Six Sigma) Американського товариства якості;
- ядро знань з сертифікації фахівця в галузі якості Американського товариства якості.

Ядра знань в області безпеки включають:

- ядро знань для фахівців у галузі безпеки;
- ядро знань Ради з практики і стандартів Американського товариства інженерів з безпеки.

Також існує багато зібраних ядер знань у галузі комп'ютерних наук та програмної інженерії, доступних для фахівців даних галузей:

- ядро знань з програмної інженерії (Software Engineering Body of knowledge - SWEBOK);
- ядро знань з індивідуальної організації процесу розробки програмного забезпечення (ПЗ);
- ядро знань з вимірювання якості програмного забезпечення;
- ядро знань щодо забезпечення гарантій ПЗ;
- ядро знань для кваліфікованого тестувальника ПЗ;
- ядро знань про техніки читання ПЗ;
- ядро знань комп'ютерних наук.

У системній інженерії можна назвати наступні ядра знань:

- ядро знань з системної інженерії;
- ядро знань з інженерії інформаційних систем;

- ядро знань для фахівців в інженерії безпеки інформаційних систем.

Той факт, що у деяких дисциплінах є кілька ядер знань, є ознакою того, що моделювання, яке має кілька аспектів, може потребувати більше ніж одне ядро знань.

Поняття моделювання вже розглядалося в даній главі. Визначення ж поняття імітаційного моделювання має два напрямки. Як процес термін імітаційне моделювання має технічне і нетехнічне значення. Як нетехнічний термін моделювання означає імітацію.

В якості технічного терміна імітаційне моделювання має два аспекти, тобто, набуття досвіду та проведення експериментів. З точки зору експериментування, імітаційне моделювання – це спрямоване на мету експериментування з використанням динамічних моделей. З точки зору досвіду, моделювання – це метод набуття досвіду в контрольованих умовах.

Технічні значення, застосовувані з 1950-х років, охоплюють будь-який тип імітаційного моделювання, незважаючи на те, комп'ютеризовано воно чи ні і чи виконано воно лише на програмному забезпеченні або за допомогою апаратного та програмного забезпечення. Крім того, обидва ці технічні визначення полегшують спадну декомпозицію сутностей і діяльностей, залучених у даний процес.

Моделювання та імітаційне моделювання, зокрема, має безліч сфер застосування: навчання, освіта, підтримка прийняття рішень, сфера розваг. З точки зору експериментування, прикладні області включають всі інженерні, технічні області, а також більшість нетехнічних областей, таких як мистецтво, біологія, хімія, управління, медицина і фізика, а також освіта і сфера розваг.

Наприклад, у військовій сфері найбільш поширеними є такі застосування моделювання: військові ігри, протиракетна оборона, комунікація та комп'ютери, розробка програм польотів, візуалізація, динамічне відображення руху, самонаведення, детекторні пристрої та обробка сигналів.

Існує два типи імітаційного моделювання, залежно від того, чи працює програма симулятор незалежно від системи, яку вона представляє, чи ні. Звідси виділяються автономні та інтегровані процеси моделювання.

Автономний процес моделювання – це процес, в якому моделююча програма працює незалежно від досліджуваної системи. Таке моделювання використовується в наступних цілях:

- тренування і удосконалення рухових навичок, навичок у прийнятті рішень і комунікаціях, операційних навичок через набуття практичного досвіду в регульованому середовищі;
- здобуття освіти;

- підтримки прийняття рішень;
- сфери розваг (імітаційні ігри, анімація динамічних систем).

Наприклад, моделювання при прийнятті рішень застосовується для здійснення наступного:

- передбачення поведінки системи та її продуктивності з обмеженнями, властивими імітаційній моделі;
- оцінка альтернативних моделей, параметрів, експериментальних і операційних умов з поведінки моделі;
- аналіз чутливості;
- технічне проектування;
- макетування;
- планування;
- збір даних;
- доказ певної концепції.

Інтегрований процес моделювання припускає, що програма моделювання працює спільно з досліджуваної системою. Основні цілі цього полягають у тому, щоб підтримати і збагатити, розширити функціонування реальної системи. Щоб підтримати функціонування реальної системи, дана система і імітаційна програма працюють поперемінно, щоб забезпечити прогнозовані результати. Для розширення функціонування реальної системи ця система і імітаційна програма працюють одночасно для забезпечення оперативної діагностики та внесення додаткової функціональності.

Моделювання може розглядатися з різних точок зору, тобто:

- інфраструктура для підтримки діяльності в реальному світі;
- обчислювальна діяльність;
- системна діяльність;
- діяльність, що базується на моделі;
- діяльність з формування знань;
- діяльність з обробки знань.

Моделювання як інфраструктура для підтримки діяльності в реальному світі відповідає сприйняттю типу чорного ящика. У цьому випадку фахівці розглядають моделювання як інструмент для досягнення інших цілей. Ця точка зору дозволяє зосередитися на первісній проблемі, з якою вони стикаються. У цьому сенсі моделювання сприймається як нереальна, неіснуюча річ.

При розумінні моделювання як обчислювальної діяльності акцент робиться на різні рівні: від відтворення моделі поведінки до використання спеціальних середовищ, які дозволяють вирішувати завдання за допомогою імітаційного моделювання. Ця точка зору відображена у деяких визначеннях моделювання. Наприклад, "Моделювання – це створення моделей, що уявляють собою атрибути однієї або декількох сутностей або процесів".

Моделювання як системна діяльність використовується для знаходження значень виходу, входу або змінних стану системи за умови, що значення двох інших видів змінних відомі. Теорія систем надає базу для формалізмів моделювання, а також для символічної обробки моделей для великої кількості динамічних систем, включаючи цілеспрямовані системи, системи зі змінною структурою, еволюційні системи.

Моделювання як діяльність, що ґрунтується на моделі, дозволяє створювати автоматизовані середовища розв’язання задач за допомогою імітаційного моделювання.

Моделювання як діяльність з формування знань передбачає, що це є цілеспрямована діяльність, яка базується на динамічних моделях. Ця точка зору дозволяє досвідченим методологам і технологам об’єднувати моделювання з кількома іншими методами здобування знань.

Моделювання як діяльність з обробки знань дозволяє методологам і технологам об’єднувати моделювання з іншими методами обробки знань.

Можна виділити різні групи зацікавлених осіб відносно можливостей, що надаються моделюванням. Не викликає сумнівів те, що різні люди можуть цікавитися різними аспектами моделювання. З прагматичної точки зору, M&SBOK повинен описувати важливі питання моделювання та імітаційного моделювання, зокрема, у всіх прикладних областях. Ці знання можуть використовуватися для декількох категорій застосування моделювання:

4) підготовка кадрів з наданням рекомендацій для розробки навчальної програми, для переддипломної та аспірантської освіти, а також для професійного розвитку;

5) основа для професійної сертифікації;

6) в якості рекомендацій для нинішніх і майбутніх професіоналів у сфері моделювання з метою планування кар’єри і самооцінки.

Будь-яка людина може скористатися M&SBOK для вивчення дисципліни моделювання та визначення того, чи потрібні йому ці знання. Фахівці-практики можуть розв’язувати певні задачі за допомогою M&SBOK. Студенти можуть розширювати і доповнювати свої знання, ґрунтуючись на порадах і рекомендаціях, що надаються M&SBOK. Викладачі вищих навчальних закладів керуються M&SBOK при розробленні навчальних планів, програм і методичного забезпечення для підготовки фахівців інженерних спеціальностей. Підприємства та представники промисловості можуть проводити атестацію своїх співробітників відносно знань, що містяться у M&SBOK. Агентства з ліцензування та сертифікації можуть керуватися M&SBOK у своїй діяльності, пов’язаній з визначенням професійних стандартів для академічних програм і комерційних організацій.

У загальному випадку фахівець в області моделювання повинен володіти трьома типами знань, а саме:

7) знання галузі застосування;
8) знання в моделюванні, на які будуть посилатися як на основні елементи;
9) знання суміжних областей.

Структура M&SBOK включає чотири логічні частини. Перша частина M&SBOK присвячена стислим попереднім питанням, передумовам створення ядра знань з моделювання. Друга частина описує ключові галузі моделювання та імітаційного моделювання. Третя частина M&SBOK пов’язана з суміжними областями знань для моделювання. В останній четвертій частині пропонуються корисні посилання в області моделювання, що включають портали, спільноти, блоги, присвячені моделюванню.

Ключові області моделювання повинні розроблятися окремо, отримуючи користь з критичного розгляду як інших досліджень у рамках M&SBOK, так і з таксономій і онтологій моделювання. До ключових областей належать наступні:

- наука / методологія;
- види імітаційного моделювання;
- життєві цикли моделювання;
- технологія;
- інфраструктура;
- надійність;
- етика;
- історія;
- напрямки, перспективи, бажані характеристики;
- підприємства;
- зрілість.

Деякі з цих областей становлять особливий інтерес при вивченні теорії моделювання. Наприклад, підрозділи такої області M&SBOK як наука/методологія, що є основами моделювання, описані в табл. 1.1.

Область, що стосується технологій, містить мови моделювання та імітаційного моделювання, засоби та відповідні середовища. Область M&SBOK про надійність охоплює питання, пов’язані з гарантуванням якості процесу моделювання і типами помилок, валідацією та верифікацією.

Таблиця 1.1 – Підрозділи області наука/методологія

№	Підрозділ	Опис
1	Дані	• Проблеми; • типи змінних; • типи значень
2	Моделі	• Проблеми, пов'язані з повторним використанням, операційною сумісністю, сполучуваністю, динамічною сполучуваністю; • концептуальні моделі і концептуальне моделювання; • таксономія імітаційних моделей; • формалізми моделювання; • моделювання фізичних систем; • моделювання якісних систем
3	Побудова моделі	• Моделювання; • побудова моделі
4	Управління, що базується на моделях	• Пошук за моделлю; • семантичний пошук за моделлю; • цілісність моделі
5	Параметри моделі і управління, що базується на параметрах	• Параметри; • допоміжні параметри; • детерміновані параметри; • стохастичні параметри
6	Характеристика моделі	• Документування моделі; • зручність використання моделі; • пояснюваність моделі
7	Оцінка моделі	• Оцінка статичної структури; • оцінка динамічної структури
8	Порівняння моделі	• Структурне порівняння моделей: • верифікація моделі (типи верифікації, методи і засоби верифікації моделі); • перевірка моделі; • еквівалентність моделей. • Поведінкове порівняння моделей (при однакових і різних сценаріях)
9	Специфікація технічної системи	• Придатність моделі (правдоподібність моделі, її адекватність, аналіз правильності моделі); • валідність моделі

Продовження таблиці 1.1

№	Підрозділ	Опис
10	Проведення експериментів	• Статистичне планування експериментів; • автоматизовані системи планування експериментів; • автоматизовані системи проведення експериментів; • методи компресії даних (детерміновані, стохастичні); • аналіз даних, отриманих у результаті імітаційного моделювання
11	Поведінка моделі	• Види поведінки моделі; • відтворення поведінки моделі; • обробка поведінки моделі; • візуалізація поведінки моделі

Ядро знань M&SBOK визначає суміжні області відносно ключових областей моделювання. До них належать області автоматизованих обчислень, допоміжні наукові, інженерні області, а також області, пов'язані з управлінням та освітою.

Обчислювальна допоміжна область розглядає цифрові, аналогові, гібридні, мобільні, хмарні обчислення, нечітку логіку, нейронні мережі. Також дана допоміжна область включає питання, пов'язані з моделями, що ґрунтуються на агентах. Наукові суміжні області включають системний аналіз, фізику, математику, теорію масового обслуговування. Серед інженерних областей можна виділити системну інженерію та візуалізацію. Управління підприємствами, проектами та продуктами відносять до суміжних областей управління.

Ядро знань M&SBOK надає корисне керівництво для професіоналів в області програмної інженерії. Оскільки моделювання і, зокрема, імітаційне моделювання являють собою важливий крок розв'язання будь-якої прикладної інженерної задачі, знання основ моделювання є актуальними для фахівців у галузі інформаційних технологій. M&SBOK може бути вкрай корисним для комплексного та інтегрованого розуміння дисципліни, для самооцінки, для атестації фахівців і організацій, а також для складання навчальних планів та методичного забезпечення вищими навчальними закладами.

Висновки

Моделювання являє значущу частину будь-якої інженерної діяльності. Моделювання дозволяє досліджувати складні об'єкти і системи, визначати їх характеристики і передбачати їх поведінку. У даній главі розглянуто поняття моделі та моделювання. Класифікація математичних моделей включає ті моделі, які будуть детально розглянуті в наступних розділах. У цьому розділі було запропоновано огляд основних положень, закріплених в ядрі знань з моделювання M&SBOK, яке розробляється для того, щоб забезпечити фахівців інженерних спеціальностей загальним розумінням основ моделювання.

Завдання

1. Розглянемо в якості досліджуваного об'єкта комутатор локальної мережі. Опишіть функціонування даного об'єкта. Визначте особливості його функціонування (неперервність або дискретність переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання та тип моделювання даного об'єкта.
2. Візьмемо в якості досліджуваного об'єкта арифметично-логічний пристрій комп'ютера. Опишіть функціонування даного об'єкта. Визначте особливості його функціонування (неперервність або дискретність переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання та тип моделювання даного об'єкта.
3. Розглянемо в якості досліджуваного явища поширення епідемії. Опишіть дане явище. Визначте особливості його протікання (неперервність або дискретність переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання та тип моделювання даного явища.
4. Розглянемо в якості досліджуваного об'єкта комірки пам'яті комп'ютера. Опишіть функціонування даного об'єкта. Визначте особливості його функціонування (неперервність або дискретність переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання та тип моделювання даного об'єкта.
5. В якості досліджуваного об'єкта подана система громадського транспорту. Опишіть функціонування даного об'єкта. Визначте особливості його функціонування (неперервність або дискретність переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання та тип моделювання даного об'єкта.
6. Розглянемо в якості досліджуваного об'єкта підприємство роздрібної торгівлі. Опишіть функціонування даного об'єкта. Визначте особливості його функціонування (неперервність або дискретність

переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання та тип моделювання даного об’єкта.

7. В якості досліджуваного процесу розглянемо боротьбу між хижаками і жертвами у природі. Опишіть даний процес. Визначте особливості його протікання (неперервність або дискретність переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання та тип моделювання для даного процесу.
8. Розглянемо в якості досліджуваного об’єкта центр обробки телефонних дзвінків. Опишіть функціонування даного об’єкта. Визначте особливості його функціонування (неперервність або дискретність переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання. Визначте тип моделювання даного об’єкта.
9. Пропонується дослідити процес обслуговування клієнтів у банку. Опишіть даний процес. Визначте особливості його протікання (неперервність або дискретність переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання та тип моделювання для даного процесу.
10. Подана в якості досліджуваного об’єкта проста телефонна система. Опишіть функціонування даного об’єкта. Визначте особливості його функціонування (неперервність або дискретність переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання та тип моделювання даного об’єкта.
11. Розглянемо в якості досліджуваного об’єкта турнікет на футбольному стадіоні. Опишіть функціонування даного об’єкта. Визначте особливості його функціонування (неперервність або дискретність переходу станів, наявність або відсутність випадкових факторів). Визначте цілі моделювання та тип моделювання даного об’єкта.

Контрольні питання

1. З якою метою використовується моделювання?
2. Дайте визначення поняттю моделі.
3. Дайте визначення поняттю моделювання.
4. Наведіть приклади моделей з навколишнього світу.
5. Чому моделі використовуються в повсякденному житті?
6. З якими видами моделей ви стикаєтеся щодня?
7. Які існують види моделей?
8. Яка основна мета моделювання?
9. Навіщо люди використовують різні типи моделей?
10. Які можна виділити цілі моделювання?

11. Дайте власне визначення поняттю моделювання.
12. Наведіть приклади застосування моделювання для вивчення нового об’єкта чи явища.
13. Наведіть приклади застосування моделювання з повсякденного життя.
14. Наведіть приклади застосування моделювання в освітньому процесі.
15. Які основні види моделей ви знаєте?
16. Дайте визначення статичній моделі.
17. Дайте визначення динамічній моделі.
18. Дайте визначення детермінованій моделі.
19. Дайте визначення стохастичній моделі.
20. Дайте визначення дискретній моделі.
21. Дайте визначення неперервній моделі.
22. Дайте визначення динамічній системі.
23. Які типи динамічних систем ви знаєте?
24. Що таке математична схема?
25. Які типи математичних схем вам відомі?
26. У яких випадках використовуються динамічні моделі?
27. Які вимоги висуваються до моделей складних систем?
28. У чому полягають кроки отримання математичних моделей елементів системи на основі нефункціональних методів?
29. Якими є цілі створення M&SBOK?
30. Які ключові галузі моделювання виділяються в M&SBOK?

2. АНАЛІТИЧНЕ МОДЕЛЮВАННЯ ОСНОВНИХ ВИДІВ ПРОЦЕСІВ У СКЛАДНИХ СИСТЕМАХ

2.1. Типові математичні схеми аналітичних моделей основних видів процесів

Розроблення аналітичної моделі реалізується на основі математичної схеми, під якою розуміється апарат чи засіб математики, що дозволяє найбільш повно відобразити характер досліджуваного процесу [6]. Прикладами математичних схем є диференціальні рівняння, скінченно-різницеві рівняння та ін.

До основних видів процесів у складних системах відповідно до характеру їх перебігу відносять неперервні, дискретні, детерміновані, стохастичні. Найбільш часто об'єктами дослідження є поєднання зазначених основних видів: *неперервно-детерміновані*, *дискретно-детерміновані*, *дискретно-стохастичні* і *неперервно-стохастичні* процеси. Для розроблення аналітичних моделей цих процесів використовуються відповідні математичні схеми, що визначаються як типові математичні схеми.

Для розроблення аналітичної моделі неперервно-детермінованих процесів в якості типової математичної схеми використовуються диференціальні рівняння; для дискретно-детермінованих – скінченні автомати; для дискретно-стохастичних – імовірнісні автомати; для неперервно-стохастичних – системи масового обслуговування.

Проблем концептуального характеру не виникає, коли всі процеси в системі одного виду, тобто всі, наприклад, неперервно-детерміновані або дискретно-детерміновані. У такому випадку модель системи подається системою моделей, кожна з яких відображає відповідний процес і розроблена на основі певної типової математичної схеми, загальної для всіх моделей. Таким чином, якщо в системі протікають кілька неперервно-детермінованих процесів, то їх моделлю може бути система диференціальних рівнянь.

Додаткові складнощі виникають, коли в системі протікають кілька процесів різного виду. Складнощі обумовлюються тим, що моделі процесів різного виду, як правило, не сполучаються в систему моделей.

Подолання зазначеної складності призвело до розроблення уніфікованої математичної схеми, що дозволяє одноманітно відображати різні види процесів. Групою вчених під керівництвом професора Бусленко Н.П. У 60-х роках минулого століття була запропонована уніфікована математична схема, яка названа агрегатом [14]. Всі раніше зазначені типові математичні схеми є окремими випадками агрегату і можуть бути отримані з нього шляхом введення певних обмежень або умов.

2.2. Неперервно-детерміноване моделювання

Для розроблення неперервно-детермінованих моделей використовується така типова математична схема, як диференціальні рівняння.

Добре відомо, що диференціальними є рівняння, до складу яких входять шукана функція однієї або декількох змінних і її похідні різних порядків. Якщо шуканою є функція однієї змінної, то диференціальне рівняння є звичайним. Якщо ж шуканою є функція декількох змінних (двох і більше), то диференціальне рівняння є рівнянням у часткових похідних.

В англомовній літературі неперервно-детерміновані моделі часто називають D-схемами (Dynamic), тим самим підкреслюючи, що цей вид моделей призначається для вивчення динаміки неперервно-детермінованих процесів, тобто для вивчення змін характеристик цих процесів у часі.

Для отримання неперервно-детермінованої моделі у формі диференціального рівняння або системи диференціальних рівнянь необхідно, по-перше, визначити природу досліджуваних процесів, яка може бути механічною, електричною, термодинамічною і т.д., і, по-друге, знати і вміти використовувати закони, за допомогою яких пояснюється перебіг процесу.

Зауважимо, що зазначені дії, пов’язані з визначенням природи процесів і законів, яким вони підпорядковуються, необхідні для отримання не тільки неперервно-детермінованих, а й інших видів моделей.

Розглянемо застосування цих дій для побудови неперервно-детермінованих моделей процесів функціонування деяких конкретних систем.

Нехай є пружинний маятник масою m, яка рухається, торкаючись стінок циліндра, що показано на рис. 2.1. Як і у випадку пружинного маятника на рис. 1.5, процес руху маси m має неперервно-детермінований характер і пояснюється законами механіки: другим законом Ньютона, законом Гука і законами тертя маси m об стінки циліндра.

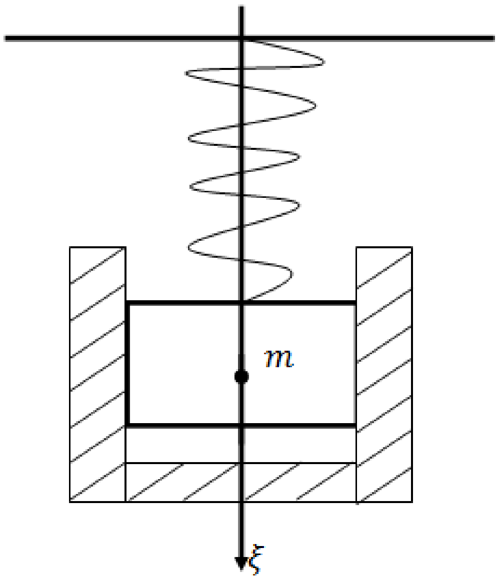


Рисунок 2.1 – Вигляд механічної системи «пружинний маятник у циліндрі»

Відповідно до другого закону Ньютона

$$a \cdot m = F,$$

де a – прискорення маси m; F – сила, що діє на масу, яка складається з двох сил –

$$F = F_1 + F_2.$$

Сила F_1 є силою розтягування–стиснення пружини, яка відповідно до закону Гука залежить від величини відхилення ξ маси m від положення рівноваги і спрямована проти руху, тобто

$$F_1 = -k_1 \xi.$$

Знак «–» у виразі сили F_1 відображає протилежність сили напрямку руху, k_1 – коефіцієнт жорсткості пружини.

Сила F_2 є силою тертя, яка пропорційна коефіцієнту k_2 швидкості руху вздовж осі $\xi - \dfrac{d\xi}{dt}$ і спрямована проти руху.

Тому

$$F_2 = -k_2 \frac{d\xi}{dt}.$$

Беручи до уваги, що прискорення дорівнює $a = \frac{d^2\xi}{dt^2}$, отримуємо остаточний вираз аналітичної моделі руху пружинного маятника в циліндрі

$$m\frac{d^2\xi}{dt^2} = -k_2 \frac{d\xi}{dt} - k_1\xi.$$

Часто цю модель подають у вигляді:

$$\frac{d^2\xi}{dt^2} + 2r\frac{d\xi}{dt} + \omega^2\xi = 0, \qquad (2.1)$$

де $2r = \frac{k_2}{m}$; $\omega^2 = \frac{k_1}{m}$.

У більш узагальненому вигляді модель (2.1) записується як

$$\frac{d^2\xi}{dt^2} = f\left(\xi, \dot{\xi}\right).$$

Аналітичні моделі у вигляді рівнянь

$$\frac{d^2\xi}{dt^2} = f(\xi); \quad \frac{d^2\xi}{dt^2} = f\left(\xi, \dot{\xi}\right),$$

праві частини яких в явному вигляді не залежать від часу *t*, відображають рух стаціонарних динамічних систем з одним ступенем свободи.

Характеристики стаціонарних динамічних систем явно не залежать від часу *t*. Зокрема, для розглянутого прикладу динамічної системи постійними є маса *m*, коефіцієнт жорсткості пружини k_1 та коефіцієнт тертя k_2.

Реальні динамічні системи, як правило, є нестаціонарними і описуються аналітичними моделями у вигляді диференціальних рівнянь, праві частини яких залежать явно від часу *t*,

$$\frac{d^2\xi}{dt^2} = f\left(t, \xi, \dot{\xi}\right).$$

Характеристики нестаціонарних динамічних систем є функціями часу. У розглянутому прикладі $m = m(t); k_1 = k_1(t); k_2 = k_2(t)$. Тілом змінної маси є ракета. Її маса змінюється в міру витрати палива у процесі польоту. Залежність $k_1 = k_1(t)$ означає, що з плином часу жорсткість пружини змінюється. Очевидно, що і коефіцієнт тертя $k_2 = k_2(t)$ змінюється в часі. Роз’яснення вимагає ще поняття «ступінь свободи». Система володіє одним ступенем свободи, якщо її можна подати у вигляді матеріальної точки, яка рухається уздовж однієї з координат простору. Якщо точка рухається в площині і її положення задається двома координатами ξ_1, ξ_2, то система володіє двома ступенями свободи. Далі, якщо положення точки визначається трьома координатами ξ_1, ξ_2, ξ_3, то система має три ступені свободи. У разі якщо система подається у вигляді n матеріальних точок, які рухаються у 3-вимірному просторі, то вона має $3n$ ступенів свободи. Для завдання положення або, в більш широкому сенсі, стану динамічних систем важливе значення має поняття простору.

Простір, у якому описуються переміщення рухомих точок, якими подається система, називається конфігураційним. Його розмірність дорівнює числу ступенів свободи динамічної системи. Миттєве положення (стан) рухомих точок називають конфігурацією динамічної системи в даний момент часу.

Розглянута як приклад динаміка руху маси пружинного маятника в циліндрі є динамікою системи, представленої однією точкою, в якій зосереджена вся маса m. Ця точка рухається вздовж осі 0ξ і тому дана система є системою з одним ступенем свободи. Положення такої системи цілком визначається в одновимірному конфігураційному просторі з віссю 0ξ. Додаючи до цієї осі ще вісь часу і поставивши у відповідність кожному моменту часу t значення ξ, отримуємо добре відомий графік функції $\xi(t)$, який показано на рис. 2.2.

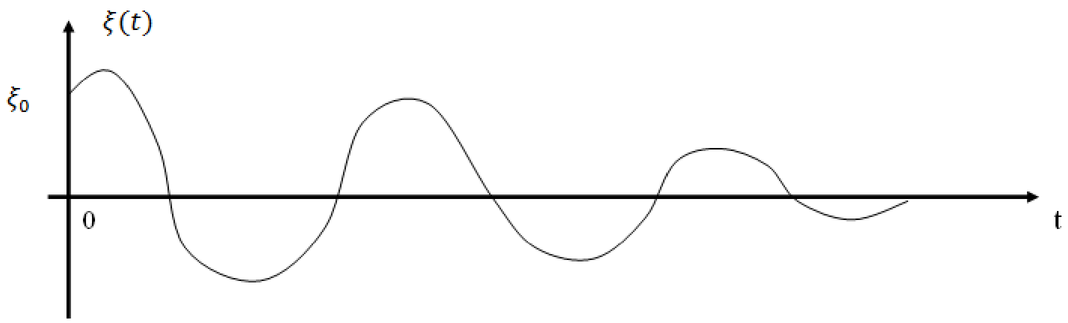


Рисунок 2.2 – Графік функції $\xi(t)$

Функція $\xi(t)$ є рішенням диференціального рівняння (2.1) і відображає затухаючі коливання з частотою ω. На графіку ξ_0 – це початкове в момент $t = 0$ відхилення маси m від положення рівноваги.

Для дослідження динамічних систем, крім конфігураційного, часто використовується простір станів. Найбільш часто простір станів використовується в дослідженнях динамічних систем методами теорії автоматичного управління [15].

У просторі станів кожна з точок $i \in \overline{1,n}$, якими подається досліджувана система, має дві координати – координату положення $z_{1i} = \xi$ і координату швидкості зміни положення $z_{2i} = \dot{\xi}$.

При русі n точок у тривимірному просторі, тобто коли динамічна система має $3n$ ступенів свободи, відповідний простір станів буде $6n$-мірним. Вектор стану такої системи має $6n$ компонент – $z = \left(z_{11}^1, z_{11}^2, z_{11}^3, z_{21}^1, z_{21}^2, z_{21}^3, \ldots, z_{1i}^1, z_{1i}^2, z_{1i}^3, z_{2i}^1, z_{2i}^2, z_{2i}^3, \ldots, z_{1n}^1, z_{1n}^2, z_{1n}^3, z_{2n}^1, z_{2n}^2, z_{2n}^3\right)$.

Компоненти $z_{1i}^1, z_{1i}^2, z_{1i}^3 \left(i \in \overline{1,n}\right)$ визначають положення i-ої точки відповідно за першою, другою, третьою координатами тривимірного простору, а компоненти $z_{2i}^1, z_{2i}^2, z_{2i}^3$ визначають швидкість зміни положення i-ї точки відповідно за першою, другою і третьою координатами.

Сукупність векторів z чи інакше точок у $6n$-вимірному просторі для всіх значень моментів часу $t \in [0, T]$ визначає траєкторію руху динамічної системи у просторі станів.

Упорядковану пару (t, z) називають подією або фазою процесу руху системи, а простір, точками якого є всілякі події, – фазовим простором динамічної системи.

Траєкторії динамічної системи у просторі станів утворюють портрет системи в просторі станів чи інакше фазовий портрет.

Отримаємо модель пружинного маятника на рис. 1.5 у просторі станів і побудуємо його фазовий портрет.

Моделлю пружинного маятника на рис. 1.5 при його русі вздовж осі $0X$ є раніше отримане звичайне диференціальне рівняння другого порядку вигляду (без урахування тертя)

$$\frac{d^2\xi}{dt^2} + \omega^2 \xi(t) = 0; \quad (k_2 = 0), \quad (\xi = x). \tag{2.2}$$

Пружинний маятник за умови, що його маса уявляється точкою, яка рухається вздовж осі 0ξ, є динамічною системою з одним ступенем свободи. Положення такої системи визначається координатою ξ, а швидкість зміни положення – першою похідною – $\frac{d\xi}{dt}$. Тому

$$z_1 = \xi,\; z_2 = \frac{d\xi}{dt} = \dot{\xi}. \tag{2.3}$$

Виходячи з введених координат простору станів z_1 і z_2, модель (2.2) перетворюється в модель у вигляді системи з двох звичайних диференціальних рівнянь першого порядку:

$$\frac{dz_1}{dt} = z_2,$$
$$\frac{dz_2}{dt} = -\omega^2 z_1. \tag{2.4}$$

Отримана модель (2.4) і є моделлю пружинного маятника у просторі станів.

Для побудови фазового портрету пружинного маятника необхідно визначити $z_1(t)$ і $z_2(t)$. Для цього необхідно розв'язати систему (2.4) або рівняння (2.2). Розв'язанням рівняння (2.2) є

$$\xi(t) = \xi_0 \cos \omega t + \frac{\dot{\xi}_0}{\omega} \sin \omega t,$$

де $\xi_0 = \xi(t_0)$, $\dot{\xi}_0 = \left.\frac{d\xi}{dt}\right|_{t=t_0}$ – початкові умови для розв'язання (2.2).

Беручи до уваги позначення (2.3), отримуємо:

$$z_1(t) = \xi(t) = \xi_0 \cos \omega t + \frac{\dot{\xi}_0}{\omega} \sin \omega t;$$
$$z_2(t) = \frac{d\xi}{dt} = -\xi_0 \omega \sin \omega t + \dot{\xi}_0 \cos \omega t. \tag{2.5}$$

Звівши у квадрат ліву і праву частини співвідношень (2.5), а потім склавши їх при $\omega = 1$, отримуємо:

$$z_1^2(t) + z_2^2(t) = \xi_0^2 + \dot{\xi}_0^2. \tag{2.6}$$

Очевидно, що (2.6) є рівнянням кола з радіусом $r_0 = \sqrt{\xi_0^2 + \dot{\xi}_0^2}$. Змінюючи значення початкових умов $\xi_0, \dot{\xi}_0$, отримуємо сімейство кіл. Це сімейство і є фазовим портретом пружинного маятника, який показано на рис. 2.3.

На рис. 2.3 подано зображуючу точку $z = (z_1, z_2)$, яка при зростанні t>0 рухається по колу в напрямку годинникової стрілки з кутовою швидкістю $\omega = 1$. Початок координат відповідає стану спокою і є виродженою окружністю при початкових умовах $\xi_0 = 0, \dot{\xi}_0 = 0$.

Фазовий портрет дозволяє встановити властивості досліджуваної системи. Дійсно, у розглянутій системі $z_1^2(t)$ пропорційно потенційній енергії, а $z_2^2(t)$ – кінетичній енергії. Сума $z_1^2(t) + z_2^2(t)$ пропорційна, очевидно, повній енергії системи, яка в даному випадку є постійною. У процесі руху один вид енергії зростає, а інший – зменшується так, що їх сума постійна.

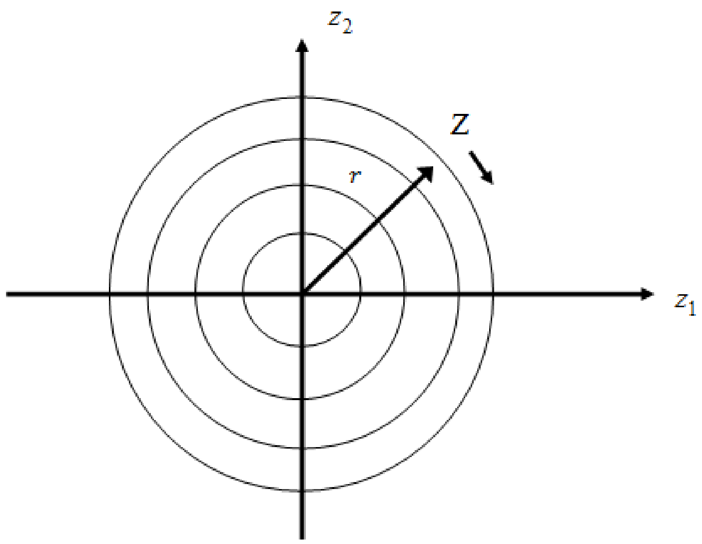


Рисунок 2.3 – Фазовий портрет пружинного маятника

Другим прикладом об'єкта моделювання може бути об'єкт іншої фізичної природи – електричний коливальний контур (LC-ланцюг) із зовнішнім джерелом напруги (рис. 2.4).

Рисунок 2.4 – Схема LC-ланцюга із зовнішнім джерелом напруги

Процеси в даному об'єкті належать до класу неперервно-детермінованих. Тому в якості математичної схеми для побудови аналітичної моделі можуть бути використані диференціальні рівняння.

Виходячи з електричної природи процесів і законів Кірхгофа, яким підкоряються процеси, отримуємо наступну аналітичну модель динаміки процесів у вигляді системи звичайних диференціальних рівнянь:

$$C\frac{dV_C}{dt} = -i_L;$$

$$L\frac{di_L}{dt} = -V_C + V_{\sim},$$

де V_C – напруга на конденсаторі ємністю *C*, i_L – струм в котушці *L*, $V_{\sim}$ – напруга зовнішнього джерела живлення.

Якщо використовувати позначення для подання результатів моделювання в просторі станів $z_1 = V_C$, $z_2 = i_L$, $x(t) = \frac{V_{\sim}}{L}$, то отримуємо модель в наступному вигляді:

$$\begin{aligned} \frac{dz_1}{dt} &= -\frac{1}{C} z_2; \\ \frac{dz_2}{dt} &= -\frac{1}{L} z_1 + x(t). \end{aligned} \tag{2.5}$$

Якщо виключити вплив зовнішнього середовища на LC-ланцюг, тобто припустити, що $x(t) = 0$, то очевидно, що з точністю до деяких позначень модель (2.4), яка описує динаміку пружинного маятника, і модель (2.5), яка описує динаміку електричного коливального контуру, збігаються. Звідси випливає важливий висновок: незважаючи на різну фізичну природу процесів у досліджуваних об’єктах, динаміка їх поведінки однакова. Цим користуються для більш глибокого вивчення та розуміння досліджуваних процесів. Так, динаміка добре відомих процесів у гідродинамічних системах може бути використана для розуміння динаміки процесів у банківській системі.

Тепер розглянемо випадок, коли $x(t) \neq 0$, тобто LC-ланцюг піддається впливу з боку зовнішнього середовища. Якщо припустити, що $x(t) = u \sin pt$, то розв’язання системи (2.5) буде

$$z_1 = V_C(t) = V_0 \cos \omega t + \frac{i_0}{\omega} \sin \omega t + \frac{up}{\omega(\omega^2 - p^2)} \sin \omega t + \frac{u}{\omega(\omega^2 - p^2)} \sin pt, \tag{2.6}$$

де $\omega = \frac{1}{\sqrt{LC}}$.

У розв’язку (2.6) є члени, які залежать від параметрів зовнішнього впливу *u, p*. Очевидно, що ці параметри можна використовувати для зміни розв’язку (2.6), а значить, у більш широкому сенсі, для зміни стану досліджуваного об’єкта.

Функцію $x(t)$ у правій частині диференціального рівняння називають вхідним процесом, а її миттєве значення при $t = t'$ називають вхідним сигналом $x_{t'} = x(t')$.

Поряд з вхідним процесом розглядають вихідні процеси $y(t)$ і вихідні сигнали y_t, які визначаються як

$$y_t = g(t, z, x(t)).$$

У загальному випадку вхідні і вихідні процеси – це вектори:

$$x(t) = (x_1(t), x_2(t), \ldots, x_m(t));$$
$$y(t) = (y_1(t), y_2(t), \ldots, y_r(t)).$$

Аналітична модель динаміки процесів у системі задається наступними групами співвідношень:

а) диференціальними рівняннями руху у просторі станів:

$$\begin{aligned} \frac{dz_1}{dt} &= f_1(t, z_1, z_2, \ldots, z_n, x_1, x_2, \ldots, x_m); \\ \frac{dz_2}{dt} &= f_2(t, z_1, z_2, \ldots, z_n, x_1, x_2, \ldots, x_m); \\ &\ldots \\ \frac{dz_n}{dt} &= f_n(t, z_1, z_2, \ldots, z_n, x_1, x_2, \ldots, x_m); \end{aligned} \tag{2.7}$$

б) початковими умовами: при $t = t_{0,}\ z_1 = z_1^0,\ z_2 = z_2^0, \ldots, z_n = z_n^0;$

в) значеннями вхідного процесу на напівінтервалі $(t_0, t]$:

$$x_1(t)\Big|_{t_0}^{t},\ x_2(t)\Big|_{t_0}^{t},\ \ldots, x_m(t)\Big|_{t_0}^{t},$$

які ще називають фрагментами вхідного процесу,

г) співвідношеннями для вихідних процесів:

$$\begin{aligned} y_1(t) &= g_1(t, z_1, z_2, \ldots, z_n, x_1, x_2, \ldots, x_m); \\ y_2(t) &= g_2(t, z_1, z_2, \ldots, z_n, x_1, x_2, \ldots, x_m); \\ &\ldots \\ y_r(t) &= g_r(t, z_1, z_2, \ldots, z_n, x_1, x_2, \ldots, x_m). \end{aligned}$$

Якщо виконуються умови існування та єдності розв’язань зазначеної системи звичайних диференціальних рівнянь (2.7), то ці розв’язання можуть бути представлені в наступному вигляді:

$$\begin{aligned} z_1(t) &= \varphi_1\left(t, t_0, z^0, x(t)\Big|_{t_0}^{t}\right), \\ z_2(t) &= \varphi_2\left(t, t_0, z^0, x(t)\Big|_{t_0}^{t}\right), \\ &\ldots \\ z_n(t) &= \varphi_n\left(t, t_0, z^0, x(t)\Big|_{t_0}^{t}\right), \end{aligned}$$

де $z^0=\left(z_1^0, z_2^0,\ldots, z_n^0\right)$ – вектор початкових умов, $x(t)\big|_{t_0}^{t}=\left(x_1(t)\big|_{t_0}^{t}, x_2(t)\big|_{t_0}^{t},\ldots,x_m(t)\big|_{t_0}^{t}\right)$ – вектор значень вхідного процесу на напівінтервалі $(t_0,t]$.

У векторному вигляді розв’язання подається як

$$z(t)=\Phi\left(t,t_0,z^0,x(t)\big|_{t_0}^{t}\right).$$

Таке подання називають ще функцією переходів динамічної системи, а функцію $y(t)=G\left(t,t_0,z^0,x(t)\big|_{t_0}^{t}\right)$ називають функцією виходів динамічної системи.

Крім зазначеного загального вигляду аналітичної моделі динамічної системи як системи звичайних диференціальних рівнянь, часто використовується модель у вигляді звичайних диференціальних або їх системи з запізнілим аргументом. До такого вигляду моделі призводить постановка, наприклад, наступного завдання управління рухом транспорту в одному ряду.

Нехай *x* – напрям руху транспорту в одному ряду, що показано на рис. 2.4.

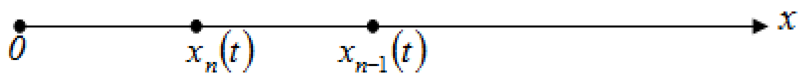


Рисунок 2.4 – Схема руху транспорту в одному ряду

Позначимо $x_n(t)$ положення *n*-го транспортного засобу (ТЗ) у момент часу *t*, а через $x_{n-1}(t)$ – положення (*n*–1) ТС, який рухається безпосередньо перед *n*-м.

Швидкості рухів *n* і *(n*–1*)* ТЗ визначаються як $\dfrac{dx_n(t)}{dt}$, $\dfrac{dx_{n-1}(t)}{dt}$.

Очевидно, що водій *n*-го ТЗ з затримкою τ реагує на зміну швидкості ТЗ, яке рухається перед ним, тобто *(n*–1*)* ТЗ. Прискорення *n*-го ТЗ подається виразом

$$\frac{d^2x_n(t+\tau)}{dt^2}=\alpha\left(\frac{dx_{n-1}(t)}{dt}-\frac{dx_n(t)}{dt}\right), \tag{2.8}$$

де α – коефіцієнт чутливості, який дорівнює константі, що має розмірність 1/с.

Позначаючи $t+\tau=\xi$, отримуємо $t=\xi-\tau, dt=d\xi$, оскільки τ є величиною постійною. З урахуванням цих позначень рівняння (2.8) приймає вигляд:

$$\frac{d^2x_n(\xi)}{d\xi^2}=\alpha\left(\frac{dx_{n-1}(\xi-\tau)}{d\xi}-\frac{dx_n(\xi-\tau)}{d\xi}\right).$$

Повертаючись до звичного позначення часу $t=\xi$, отримуємо остаточний вигляд аналітичної моделі динаміки ТЗ, що рухаються в одному ряду:

$$\frac{d^2x_n(t)}{dt^2}=\alpha\left(\frac{dx_{n-1}(t-\tau)}{dt}-\frac{dx_n(t-\tau)}{dt}\right). \quad (2.9)$$

Модель (2.9) являє собою звичайне диференціальне рівняння другого порядку з запізнілим аргументом. Модель (2.9) може бути перетворена в модель у просторі станів шляхом введення таких позначень:

$$x_n(t)=z_1; \quad \frac{dx_n}{dt}=z_2; \quad \frac{dx_{n-1}(t)}{dt}=x(t).$$

Перші два позначення звичайні для простору стану, а позначення швидкості *(n*–1*)* ТЗ як вхідного процесу означає, що *(n*–1*)*-е ТЗ є для *n*-го ТЗ "зовнішнім середовищем", зміна стану якого є вхідним процесом для розглянутого *n*-го ТЗ як досліджуваної системи.

З урахуванням введених позначень модель динаміки ТЗ, що рухаються в один ряд, у просторі станів набуває наступного вигляду:

$$\frac{dz_1}{dt}=z_2;$$
$$\frac{dz_2}{dt}=\alpha[x(t-\tau)-z_2(t-\tau)].$$

Найпростіше диференціальне рівняння з запізнілим аргументом має вигляд

$$\frac{dz(t)}{dt}=f[t,z(t),z(t-\tau)].$$

Розв’язання цього рівняння подається як

$$z(t)=z_0+\int_{t_0}^{t} f(\Theta,z(\Theta),z(\Theta-\tau))d\Theta,$$

тобто для отримання розв'язання, крім початкових умов, необхідно і достатньо задати функцію $z(t)$ в напівінтервалі $[t_0 - \tau, t_0)$ чи інакше початкову функцію.

Аналітичні моделі у вигляді диференціальних рівнянь з запізнілим аргументом використовуються для опису динамічних систем з післядією, тобто систем, для визначення стану яких $z(t)$ при $t > t_0$, крім завдання початкового стану $z_0 = z(t_0)$, необхідно задати передісторію стану у вигляді початкової функції $z(t)\big|_{t_0-\tau}^{t_0}$, тобто функцію $z(t)$ у напівінтервалі $[t_0 - \tau, t_0)$.

2.3. Моделі у вигляді скінченно-різницевих рівнянь

Скінченно-різницеві рівняння можуть використовуватися в якості математичної схеми для розроблення аналітичних моделей як неперервно-детермінованих, так і дискретно-детермінованих процесів. Дійсно, з одного боку, скінченно-різницеві рівняння є перетворенням диференціальних рівнянь, які, як показано в попередньому підрозділі, описують динаміку неперервно-детермінованих процесів. Перетворення ґрунтується на заміні похідних виразом вигляду

$$\frac{dz_i}{dt} \approx \frac{z_i(t+h) - z_i(t)}{h},$$

де h – досить мала величина, яку в чисельних методах розглядають як крок інтегрування.

Зазначене перетворення дозволяє уявити систему диференціальних рівнянь (2.7) у вигляді такої системи скінченно-різницевих рівнянь:

$$\begin{aligned} &z_1(t+h) = z_1(t) + hf_1(t, z_1, z_2, \dots, z_n, x_1, x_2, \dots, x_m); \\ &z_2(t+h) = z_2(t) + hf_2(t, z_1, z_2, \dots, z_n, x_1, x_2, \dots, x_m); \\ &\dots \\ &z_n(t+h) = z_n(t) + hf_n(t, z_1, z_2, \dots, z_n, x_1, x_2, \dots, x_m). \end{aligned} \tag{2.10}$$

Система скінченно-різницевих рівнянь може розглядатися як аналітична модель неперервно-детермінованих процесів, стан яких визначається системою диференціальних рівнянь (2.7).

З іншого боку, скінченно-різницеві рівняння є самостійною математичною схемою як основою для розроблення аналітичних моделей дискретно-детермінованих процесів. Прикладом такої моделі є модель для розв'язання макроекономічної задачі зв'язків «витрати – випуск» у рамках корпорації підприємств [16].

Нехай z_i – валовий випуск *i*-го продукту в одиницю часу; a_{ij} – кількість *i*-го продукту, спожита для виготовлення одиниці *j*-го продукту. Тоді модель самозабезпечення виробництва в рамках корпорації має вигляд:

$$\forall j \in \overline{1,n}; \quad z_j(t+1) = \sum_{i \in 1,n} a_{ij} z_i(t); \quad z(0) = z_0. \tag{2.11}$$

Очевидно, що (2.11) є системою скінченно-різницевих рівнянь вигляду (2.10) за умови лінійного характеру функції $f_i\left(i \in \overline{1,n}\right)$ та *h*=1. Інакше (2.11) називають однокроковою функцією переходів динамічної системи, що описується скінченно-різницевими рівняннями.

2.4. Дискретно-детерміноване моделювання

Основними математичними схемами для розроблення аналітичних моделей дискретно-детермінованих процесів разом із зазначеними в попередньому підрозділі скінченно-різницевими рівняннями є різного виду скінченні автомати (Finite automata).

Одним з основних понять теорії скінченних автоматів є поняття алфавіту як скінченної множини букв, кожна з яких позначає певний об’єкт. Об’єктами є значки, цифри, букви, рисунки, фрази, алгоритми і т.д.

Скінченну упорядковану сукупність букв називають словом у даному алфавіті.

Скінченний автомат Мілі або Мура забезпечує перетворення сигналів вхідного алфавіту *X* у сигнали вихідного алфавіту *Y*.

Скінченний автомат Мілі функціонує в момент часу $t_0 < t_1 < t_2 ...$ так, що в кожен момент $t_j \in T$ (*T* – множина моментів часу) він знаходиться в одному з можливих станів $z_j = z(t_j) \in Z$ (*Z* – скінченна множина станів) і після надходження вхідного сигналу $x \in X$ в момент часу $t_j \in T$, починаючи з моменту t_1, переходить у наступний стан з множини *Z* і видає вихідний сигнал $y \in Y$ [17, 18].

Використовуються три основні форми завдання автомата – аналітична, таблична і графова. Всі ці форми забезпечують перетворення, в результаті якого визначається новий стан автомата і вихідний сигнал.

Аналітична форма завдання скінченного автомата Мілі передбачає завдання однокрокової функції переходів φ для визначення нового стану скінченного автомата і функції виходів ψ для визначення вихідного сигналу:

$$z(t)=\varphi[z(t-1),x(t)];$$
$$y(t)=\psi[z(t-1),x(t)]. \qquad (2.11)$$

З (2.11) випливає, що для визначення нового стану автомата $z(t)$ і вихідного сигналу $y(t)$ в момент часу $t\in T$ необхідно знати стан автомата в попередній момент часу $z(t-1)$ і вхідний сигнал в момент $t\in T$ – $x(t)$.

Таблична форма завдання скінченного автомата Мілі передбачає завдання двох таблиць: таблиці переходів, яка відповідає функції φ, і таблиці виходів, яка відповідає функції ψ у (2.11).

Нехай, наприклад, вхідний алфавіт *X*, множини станів *Z*, вихідний алфавіт *Y* мають вигляд:

$$X=\{x_1,x_2,x_3\},\ Z=\{z_1,z_2,z_3,z_4\},\ Y=\{y_1,y_2\}.$$

Тоді таблиця переходів “φ” і таблиця виходів “ψ” можуть мати вигляд відповідно табл. 2.1 і 2.2. Структура таблиць однакова – їх рядки відповідають значенням вхідного сигналу $x(t)$ в моменти $t\in T$, а стовпці відповідають станам автомата в попередній *t*–1 момент часу $z(t-1)$.

На перетині рядків і стовпців табл. 2.1 записуються стани автомата, в які він переходить у момент $t\in T$, тобто $z(t)$. Так, наприклад, якщо автомат знаходився у момент $t-1\in T$ у стані $z(t-1)=z_1$ (перший стовпець), а в момент $t\in T$ прийшов вхідний сигнал $x(t)=x_3$ (третій рядок), то автомат перейде в момент $t\in T$ в новий стан $z(t)=z_2$, що визначається перетином першого стовпця і третього рядка.

Таблиця 2.1 – Таблиця переходів “φ”

$x(t)$	$z(t-1)$			
	z_1	z_2	z_3	z_4
x_1	z_2	z_3	z_1	z_2
x_2	z_1	z_1	z_4	z_4
x_3	z_2	z_3	z_2	z_1

Таблиця 2.2 – Таблиця виходів “ψ”

$x(t)$	$z(t-1)$			
	z_1	z_2	z_3	z_4
x_1	y_1	y_2	y_1	y_2
x_2	y_2	y_1	y_2	y_1
x_3	y_1	y_1	y_2	y_2

На перетині рядків і стовпців табл. 2.2 записуються значення вихідного сигналу $y(t) \in Y$, що видаються в момент $t \in T$. Так, якщо автомат знаходився у момент $(t–1)$ у стані $z(t-1) = z_1$, а в момент t надійшов вхідний сигнал $x(t) = x_3$, то автомат видає вихідний сигнал $y(t) = y_1$, який в таблиці встановлюється шляхом перетину першого стовпця і третього рядка.

Подання скінченного автомата Мілі у формі орієнтованого графа передбачає відображення станів автомата $z_j = z(t_j) \in Z$ вершинами графа і переходів зі стану $z_i = z(t-1)$ у стан $z_j = z(t)$ орієнтованими дугами (стрілками) (z_i, z_j). При цьому зазвичай на початку стрілки надписується вхідний сигнал $x(t)$, який викликає перехід, а в кінці стрілки надписується вихідний сигнал $y(t)$, який супроводжує перехід.

Для розглянутого прикладу автомата Мілі, переходи і вихідні сигнали якого показано в табл. 2.1 і 2.2, орієнтований граф має вигляд, поданий на рис. 2.5.

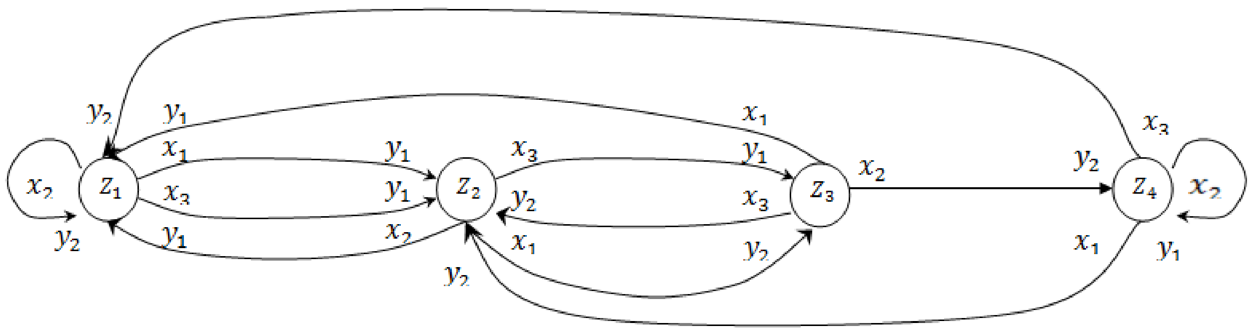


Рисунок 2.5 – Орієнтований граф автомата Мілі

Іншим видом скінченних автоматів, який часто використовується для моделювання, є автомат Мура. Цей автомат функціонує аналогічно автомату Мілі, має ті ж форми завдання і відрізняється лише вихідною функцією, яка визначається як

$$y(t) = \psi'[z(t)],$$

де ψ' називається здвинутою функцією виходів.

Більш стисло автомат Мілі часто позначається як $A(X, Z, Y, \varphi, \psi)$, а автомат Мура – як $A'(X, Z, Y, \varphi, \psi')$.

Скінченний автомат є синхронним, якщо моменти надходження вхідного сигналу $t_1, t_2 ...$ заздалегідь фіксовані і задовольняють співвідношенню $t_1 < t_2 < ...$.

Скінченний автомат є асинхронним, якщо моменти надходження вхідного сигналу $t_1, t_2...$ заздалегідь не фіксуються, а лише задовольняють співвідношенню $t_1 < t_2 < ...$. Оскільки в цьому випадку зазначені моменти часу можуть бути досить близькі один одному, то асинхронний автомат може використовуватися як математична схема для розроблення аналітичних моделей динамічних систем, що функціонують у неперервному часі.

Розглянемо деякі приклади аналітичних моделей на основі скінченних автоматів.

Приклад 1. Потрібно розробити модель функціонування пристрою з продажу квитків. Пристрій приймає жетони або монети номіналом 1, 2, 3, 5 одиниць і видає квитки вартістю 5 одиниць. Пристрій після видачі квитка обнуляє свій стан і здачі не дає.

Для побудови моделі на основі скінченного автомата Мілі або Мура необхідно визначити множину вхідних сигналів *X*, множину станів *Z*, множину вихідних сигналів *Y*, а також в тій чи іншій формі функції φ і ψ (для автомата Мура ψ').

Елементами множини *X* в аналізованому об'єкті є жетони або монети різного номіналу, які покупці можуть використовувати для придбання квитків. Тому очевидно, що $X = \{1,2,3,5\}$.

Можливими станами пристрою можуть бути: «0», коли в пристрої немає жодної монети; «1», коли у пристрої є одна монета номіналом 1; «2», коли у пристрої або одна монета номіналом 2, або дві монети номіналом 1 кожна; «3», коли у пристрої одна монета номіналом 3 або дві монети, одна з яких номіналом 1, а інша – номіналом 2; «4», коли у пристрої дві монети номіналом 1 і одна – номіналом 2 або дві монети номіналом 2, або, нарешті , дві монети, одна з яких номіналом 1, а інша – номіналом 3.

Якщо у пристрої виявляються монети сумарним номіналом 5 і більше, то згідно з умовами функціонування пристрою він видає покупцеві квиток і обнуляє стан.

Таким чином, множина можливих станів пристрою має вигляд $Z = \{0,1,2,3,4\}$. Елементами множини вихідних сигналів у розглянутому об'єкті можуть бути «видача квитка» і «квиток не видається». Якщо елементу «видача квитка» поставити у відповідність «1», а елементу «квиток не видається» – «0», то множина *Y* набуває вигляду $Y = \{0,1\}$.

Визначення функцій φ і ψ рекомендується починати з табличної форми їх завдання. Таблиці переходів "φ" та виходів "ψ" для розроблюваної моделі мають відповідно вигляд табл. 2.3 і 2.4.

Таблиця 2.3 – Таблиця переходів “φ”” у моделі функціонування пристроїв з продажу квитків

x	$z(t-1)$				
	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	0
3	3	4	0	0	0
5	0	0	0	0	0

У клітинах на перетині рядків і стовпців табл. 2.3 записуються елементи множини Z як результат підсумовування значень $z(t-1)$ і $x(t)$ з урахуванням того, що результат не більше 4. Якщо результат більше 4, то значення $z(t)$ дорівнює нулю. Звідси випливає такий вигляд аналітичної форми завдання функції φ:

$$z(t)=\begin{cases} z(t-1)+x(t), \text{ якщо } z(t-1)+x(t)\le 4; \\ 0, \qquad \text{якщо } z(t-1)+x(t)>4. \end{cases} \tag{2.12}$$

Таблиця 2.4 – Таблиця виходів “ψ” у моделі функціонування пристроїв з продажу квитків

x	$z(t-1)$				
	0	1	2	3	4
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	1	1
5	1	1	1	1	1

У клітинах табл. 2.4 записуються елементи множини Y. При цьому очевидно, що $y = 0$, коли стан пристрою визначається значеннями в межах від 1 до 4, і $y = 1$, коли стан пристрою визначається значеннями 5 і більше, яким в табл. 2.3 відповідають нулі. Таким чином, аналітична форма завдання функції ψ подається як

$$y(t)=\begin{cases} 0, \text{ якщо } z(t-1)+x(t)\le 4; \\ 1, \text{ якщо } z(t-1)+x(t)>4. \end{cases} \tag{2.13}$$

Графова форма автомата Мілі для розглянутого об’єкта моделювання має вигляд, поданий на рис. 2.6.

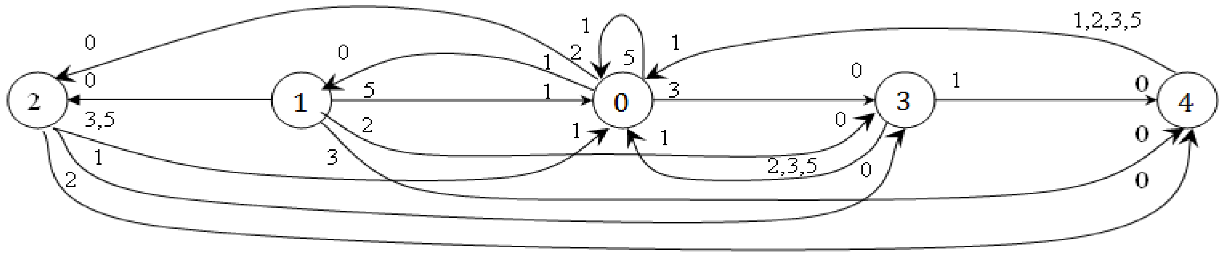


Рисунок 2.6 – Граф автомата Мілі, що моделює функціонування пристрою з продажу квитків

Таким чином, модель процесу функціонування пристрою з продажу квитків на основі автомата Мілі включає зазначені множини X, Y, Z, а також функції φ і ψ, задані або у вигляді табл. 2.3, 2.4, або у вигляді співвідношень (2.12), (2.13), або, нарешті, у вигляді графа на рис. 2.6.

Приклад 2. Потрібно розробити модель функціонування ліфта у триповерховому будинку.

Як і в попередньому прикладі, визначаємо спочатку множину вхідних сигналів X, множину станів Z і множину вихідних сигналів Y.

Елементами множини X є сигнали $x_i\left(i \in \overline{1,3}\right)$ виклику ліфта на i-й поверх, тобто $X = \left\{x_i \mid i \in \overline{1,3}\right\}$.

Множина $Z = \left\{z_i \mid i \in \overline{1,3}\right\}$, де z_i відповідає стану «ліфт знаходиться на i-му поверсі».

Множина Y може складатися з елементів, які відповідають напрямку руху ліфта. Наприклад, $Y = \left\{y_i \mid i \in \overline{1,3}\right\}$, де y_1 – «ліфт рухається вгору», y_2 – «ліфт стоїть на місці», y_3 – «ліфт рухається вниз».

Таблична форма завдання функцій φ і ψ подана у табл. 2.5 і 2.6.

Таблиця 2.5 – Таблиця переходів “φ” в моделі функціонування ліфта

x	$z(t-1)$		
	z_1	z_2	z_3
x_1	z_1	z_1	z_1
x_2	z_2	z_2	z_2
x_3	z_3	z_3	z_3

Таблиця 2.6 – Таблиця виходів “ψ” в моделі функціонування ліфта

x	$z(t-1)$		
	z_1	z_2	z_3
x_1	y_2	y_3	y_3
x_2	y_1	y_2	y_3
x_3	y_1	y_1	y_2

Значення $z(t)$ у табл. 2.5 такі, що відповідають вимогам користувача, тобто сигналам $x(t)$, незалежно від того, де знаходився ліфт у попередній $(t-1)$ момент часу.

Значення $y(t)$ в табл. 2.6 визначаються на зіставленні значень $x(t)$ та $z(t-1)$, тобто сигналу виклику в даний момент часу і стану ліфта в попередній момент часу. Наприклад, якщо в момент часу $(t-1)$ ліфт знаходиться в стані z_1 (на першому поверсі), а в момент часу t надходить вхідний сигнал $x(t)=x_1$ (виклик на перший поверх), то вихідним сигналом, очевидно, буде $y(t)=y_2$, тобто «ліфт стоїть на місці»

Аналітично функції φ і ψ визначаються виразами:

$$z(t)=x(t)\,; \tag{2.14}$$

$$y(t)=\begin{cases} y_1, \text{ якщо } x_i > z_i; \\ y_2, \text{ якщо } x_i = z_i; \\ y_3, \text{ якщо } x_i < z_i; \end{cases} \quad \left(i \in \overline{1,3}\right) \tag{2.15}$$

Графова форма завдання φ і ψ показана на рис. 2.7.

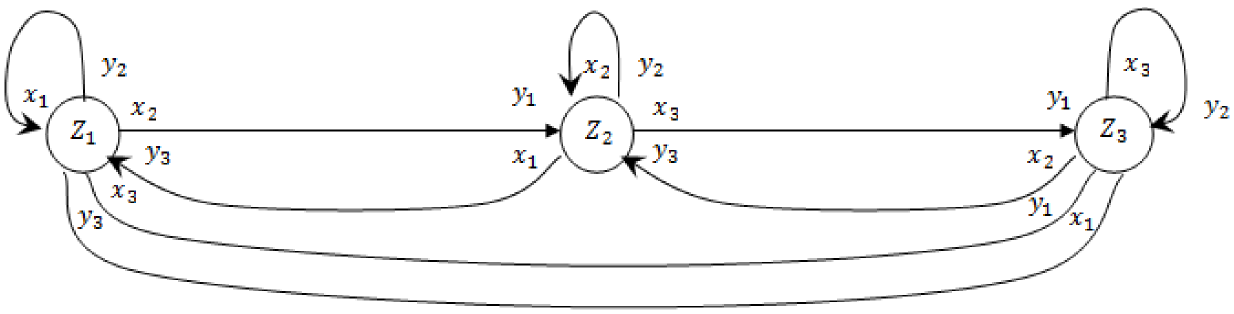


Рисунок 2.7 – Граф автомата Мілі, що моделює функціонування ліфта

Крім зазначених скінченних автоматів Мілі та Мура для побудови аналітичних моделей дискретно-детермінованих процесів часто використовуються скінченні автомати з післядією і нестаціонарні скінченні автомати.

Скінченний автомат з післядією при побудові дискретно-детермінованих моделей дозволяє врахувати передісторію досліджуваного процесу і в цьому сенсі є аналогом диференціальних рівнянь з запізнілим аргументом, що використовуються для побудови неперервно-детермінованих моделей.

Скінченний автомат з післядією визначається як

$$\begin{aligned} z(t) &= \varphi[z(t-k), z(t-k+1), \ldots, z(t-1), x(t)]; \\ y(t) &= \psi[z(t-1), x(t)], \end{aligned} \tag{2.16}$$

де k – натуральне число, що зветься порядком початкової множини.

Набір станів $z(t-k), z(t-k+1), \ldots, z(t-1)$ називається передісторією автомата, а набір моментів часу $t-k, t-k+1, \ldots, t-1$ – початковою множиною щодо моменту $t-1$.

Очевидно, що при k=1 автомат (2.16) перетворюється на звичайний скінченний автомат Мілі.

Автомат з післядією як математична схема використовується для розроблення моделей систем, для яких є характерним фактор запізнювання у процесі їх функціонування. Наприклад, віддача від вкладених інвестицій з’являється не відразу, а після деякого інтервалу часу, довжина якого якраз і враховується або значенням τ у диференціальних рівняннях з запізнілим аргументом, або значенням k у розглянутій математичній схемі автомата з післядією.

Нестаціонарний автомат дозволяє враховувати залежність змін зовнішнього середовища, структури і параметрів модельованої системи від часу. Формально ця залежність виражається у вигляді явної залежності функцій φ і ψ від часу. Таким чином, нестаціонарний автомат визначається як

$$\begin{aligned} z(t) &= \varphi[t-1, z(t-1), x(t)]; \\ y(t) &= \psi[t-1, z(t-1), x(t)]. \end{aligned} \tag{2.17}$$

Якщо час t розглядати як додаткову координату стану автомата, тобто ввести нове позначення стану як $\tilde{z}(t) = (t, z(t))$, то у співвідношеннях можна позбутися від явної залежності функцій переходів φ і виходів ψ від часу t.

Дійсно, припускаючи, що $\tilde{z}(t-1) = (t-1, z(t-1))$, отримаємо

$$\begin{aligned} \tilde{z}(t) &= \varphi[\tilde{z}(t-1), x(t)]; \\ \tilde{y}(t) &= \psi[\tilde{z}(t-1), x(t)]. \end{aligned} \tag{2.18}$$

За своїм виглядом отриманий автомат (2.18) є стаціонарним, але вже не скінченним, оскільки станів вигляду $\tilde{z}(t-1)$ стільки же, скільки моментів часу $t-1$. Однак у практиці моделювання часто на інтервалі моделювання $[0,T]$ беруть скінченну множину моментів часу $\{t\}$ і тому функціонування автомата (2.18) буде аналогічним функціонуванню звичайного скінченного автомата Мілі або Мура.

2.5. Дискретно-стохастичне моделювання

Функціонування стохастичних динамічних систем у дискретному часі часто описують за допомогою імовірнісного автомата (probabilistic automata).

Поняття імовірнісного автомата ґрунтується на єдності трьох понять – автомата з випадковими переходами, автомата з випадковим вибором початкового стану і автомата з випадковим вибором вихідного сигналу.

Автомат з випадковими переходами або з випадковою функцією переходів φ визначається сукупністю матриць $\{\|P_{ij}(x)\|\}$, елементи яких $P_{ij}(x) = P\{z(t) = z_j \,/\, z(t-1) = z_i; x(t) = x\}$ являють собою умовні ймовірності того, що в момент часу t автомат опиниться у стані z_j, тобто $z(t) = z_j$, за умови, що в попередній момент він був у стані z_i, тобто $z(t-1) = z_i$, і в момент часу t поступив зовнішній сигнал $x(t) = x$.

Автомат з випадковими переходами функціонує наступним чином. Нехай у момент t надходить вхідний сигнал $x(t) = \hat{x}$. За значенням $\hat{x}$ із сукупності матриць $\{\|P_{ij}(x)\|\}$ обирається матриця $\|P_{ij}(\hat{x})\|$.

Якщо автомат у попередній t–1 момент часу перебував у стані $z(t-1) = z_i$, то в матриці $\|P_{ij}(\hat{x})\|$ вибирається i-ий рядок зі значеннями умовних ймовірностей p_{ij}. Цей рядок має вигляд

$$p_{i1}\ p_{i2}\ \cdots\ p_{ij}\ \cdots\ p_{in},$$

а його елементи задовольняють умовам:

$$\forall i, j \quad 0 \le p_{ij} \le 1, \quad \sum_{j \in \overline{1,n}} p_{ij} = 1.$$

Зазначені умови забезпечують можливість обов'язкового переходу зі стану z_i в один зі станів z_j.

Для вибору наступного стану $z(t) = z_j$ використовується процедура жереба [19]. Відповідно до цієї процедури, з послідовності випадкових, рівномірно розподілених в інтервалі (0,1) чисел $\{\xi_k \mid k \in 1,2,\ldots\}$ вибирається випадковим чином деяке з чисел $\overline{\xi}_k$.

Якщо $0 < \overline{\xi}_k \le p_{i1}$, то $z(t) = z_1$.

Якщо $p_{i1} < \overline{\xi}_k \le p_{i1} + p_{i2}$, то $z(t) = z_2$. І так далі.

У загальному вигляді, якщо $\sum\limits_{\nu \in \overline{1, j-1}} p_{i\nu} < \overline{\xi}_k \le \sum\limits_{\nu \in \overline{1, j}} p_{i\nu}$, то $z(t) = z_j$.

Тому що $\overline{\xi}_k \in (0,1)$, число станів *n* скінченне і $\sum\limits_{j \in \overline{1,n}} p_{ij} = 1$, то зазначена процедура обов'язково забезпечує вибір конкретного стану z_j.

На наступному кроці функціонування автомата стан z_j розглядатиметься як попередній, тобто той, що відноситься до моменту $(t-1)$ – $z(t-1) = z_j$. Для визначення наступного стану $z(t)$ необхідно повторити весь описаний крок функціонування автомата.

Можна показати, що багаторазова реалізація функціонування автомата з випадковими переходами призводить до того, що частоти вибору станів $z_1, z_2, \ldots z_n$ за умови $z(t-1) = z_i$ будуть близькі до відповідних ймовірностей $p_{i1}\ p_{i2} \ldots p_{in}$.

Реалізація кожного кроку функціонування автомата з випадковими переходами зумовлюється знанням стану в попередній момент часу. Очевидно, що для реалізації першого кроку необхідно знання початкового стану або стану автомата в момент $t = 0$, тобто $z(0) = z_0$.

Якщо стан $z(0)$ не зафіксовано, а він вибирається за жеребом на основі розподілу ймовірностей $p_0 = (p_{01}, p_{02}, \ldots, p_{0n})$, де $0 \le p_{0i} \le 1 \left(i \in \overline{1,n}\right)$ – ймовірність того, що початковим станом є стан z_i, а $\sum\limits_{i \in \overline{1,n}} p_{0i} = 1$, то таку стохастичну систему називають автоматом з випадковим вибором початкового стану.

Нарешті, автомат з випадковим вибором вихідного сигналу чи автомат з випадковою функцією виходів ψ визначається сукупністю матриць $\{\|Q_{ir}(x)\|\}$, елементи котрих

$$q_{ip}(x) = P\{y(t) = y_p / z(t-1) = z_i; x(t) = x\}$$

являють собою умовні ймовірності видачі автоматом вихідного сигналу $y(t) = y_p$ в момент часу *t* за умови, що $z(t-1) = z_i$, $x(t) = x$.

Автомат з випадковим вибором вихідного сигналу функціонує аналогічно автомату з випадковими переходами.

У цілому імовірнісним автоматом є об'єкт

$$A\left(X, Z, Y, \{M(y/x)\}\right),$$

де $X = \left\{x_r \mid r \in \overline{1,l}\right\}$ – вхідний алфавіт (множина вхідних сигналів); $Z = \left\{z_i \mid i \in \overline{1,n}\right\}$ – множина станів; $Y = \left\{y_p \mid p \in \overline{1,m}\right\}$ – вихідний алфавіт (множина вихідних сигналів); $\{M(y/x)\}$ – сімейство $l \times m$ матриць, кожна з яких розмірністю $n \times n$.

У матриці $M(y_p/x_r)$ елемент $\mu_{ij}(y_p/x_r)$ визначається як умовна ймовірність

$$\mu_{ij}(y_p/x_r) = P\{z(t) = z_j, y(t) = y_p / z(t-1) = z_i, x(t) = x_r\},$$

тобто як ймовірність того, що автомат перейде в стан $z(t) = z_j$, видасть вихідний сигнал $y(t) = y_p$ за умови того, що в попередній *(t–1)* момент часу автомат знаходився в стані $z(t-1) = z_i$, а на вхід в момент часу *t* надійшов сигнал $x(t) = x_r$.

Якщо елементи $\mu_{ij}(y_p/x_r)$ приймають значення нуля або одиниці, то імовірнісний автомат перетворюється в раніше описаний кінцевий автомат. При цьому це буде кінцевий автомат Мілі за умови, що $P\{z(t) = z_j, y(t) = y_p / z(t-1) = z_i, x(t) = x_r\} = P\{z(t) = z_j / z(t-1) = z_i, x(t) = x_r\} \times$
$\times P\{y(t) = y_p / z(t-1) = z_i, x(t) = x_r\}$,
тобто випадкові величини z_j, y_p є умовно незалежними.

Якщо

$$P\{y(t) = y_p / z(t-1) = z_i, x(t) = x_r\} = P\{y(t) = y_p / z(t) = z_j\}$$

усюди, де $P\{z(t) = z_j / z(t-1) = z_i, x(t) = x_r\} \neq 0$, то кінцевий автомат є автоматом Мура.

Прикладом моделі функціонування дискретно-стохастичної системи на основі автомата з випадковими переходами є модель функціонування торгового підприємства, яке реалізує штучний товар, початковий обсяг якого заздалегідь відомий [20].

Функціонування торгового підприємства багато в чому пов'язується з керуванням товарообігом, яке реалізується на основі функції зміни обсягу продажів у часі. Побудова цієї функції можлива шляхом виконання дій з дискретно-стохастичною моделлю, яка описує процеси появи покупців та придбання ними штучних товарів, що призводить до зміни обсягу товарів у торговельному підприємстві.

Модель на основі автомата з випадковими переходами будується на основі наступних передумов і припущень.

Торгове підприємство реалізує штучні товари *n* найменувань. Його стан у кожний дискретний момент часу $t \in \{0,1,2,...\}$ визначається сукупністю обсягів товарів кожного найменування $\{z_j(t) \mid j \in \overline{1,n}\}$. При цьому

$$\forall j \in \overline{1,n} \quad \max_t z_j(t) = z_j(0) = N_j \ ;$$
$$\forall j \in \overline{1,n} \quad \min_t z_j(t) = z_j(T_j) = 0,$$

де N_j – вихідний обсяг товару *j*-го найменування, а T_j – момент часу повної реалізації початкового об’єму товару.

Далі передбачається, що зміна стану торговельного підприємства відбувається дискретно по кожній окремій координаті $z_j(t)$. Тому моделювання зміни стану проводиться для однієї координати $z_j(t)$ і при цьому індекс *j* опускається, тобто координатою стану є $z(t)$, яка задовольняє умові $0 \le z(t) \le N$. При цьому $z(0) = N,\ z(T) = 0$.

Число покупців, які відвідують торгове підприємство в момент часу *t*, починаючи з $t = 1$, є випадковою величиною $m(t)$, яка змінюється у відомих межах – $m(t) \in \overline{0, M(t)}$. Передбачається відомим розподіл ймовірностей появи того чи іншого числа покупців для кожного моменту часу, тобто

$$\forall t \in \{1,2,3,\ldots\} \quad \left\{P^t_{m(t)} \mid m(t) \in \overline{0, M(t)}\right\}, \qquad (2.19)$$

де $P^t_{m(t)}$ – ймовірність появи $m(t)$ покупців у момент часу *t*.

При цьому значення $P^t_{m(t)}$ задовольняють відомим співвідношенням:

$$\forall t \in \{1,2,3,\ldots\},\ \forall m(t) \in \overline{0, M(t)} \quad 0 \le P^t_{m(t)} \le 1,$$
$$\forall t \in \{1,2,3,\ldots\} \quad \sum_{m(t) \in \overline{0,M(t)}} P^t_{m(t)} = 1.$$

Кожен покупець з імовірністю *p* купує товар і з імовірністю *(*1*–p)* не купує його. Якщо в момент *t* в торговому підприємстві виявилося $m(t)$ покупців, то обсяг придбаного ними товару $r(t) \in \overline{0, m(t)}$, а ймовірність придбання $m(t)$ покупцями $r(t)$ одиниць штучного товару визначається як

$$P[r(t)] = C^{r(t)}_{m(t)} p^{r(t)} \cdot (1-p)^{m(t)-r(t)}, \qquad (2.20)$$

де $C^{r(t)}_{m(t)} = \dfrac{m(t)!}{r(t)!\,(m(t)-r(t))!}$.

Дискретно-стохастична модель для побудови функції зміни обсягу штучного товару з урахуванням зроблених припущень подається сукупністю матриць

$$\left\{\left\|P_{ij}(x)\right\| \mid x \in X\right\}. \qquad (2.21)$$

Вхідним сигналом *x* у пропонованій моделі є число покупців $m(t)$. Тому модель являє собою сукупність з $M(t)+1$ матриць виду (2.21) для кожного моменту часу $t \in \{1,2,3,\ldots\}$, тобто

$$\forall t \in \{1,2,3,\ldots\} \quad \left\{P_{ij}(m(t)) \mid m(t) \in \overline{0, M(t)}\right\}. \qquad (2.22)$$

Структура матриць (2.22), що визначається числом рядків і стовпців, і значення елементів $P_{ij}(m(t))$, сума яких у кожному рядку будь-якої з матриць (2.22) дорівнює одиниці, залежить від $m(t)$. Нижче наводиться вид матриць (2.22) для різних значень $m(t)$:

$$P(0) = \begin{array}{c} \\ N \\ N-1 \\ N-2 \\ \\ 2 \\ 1 \\ 0 \end{array} \begin{array}{c} \begin{array}{ccccccc} N & N-1 & N-2 & & 2 & 1 & 0 \end{array} \\ \left\| \begin{array}{ccccccc} 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 1 \end{array} \right\| \end{array},$$

$$P(1) = \begin{array}{c} \\ N \\ N-1 \\ N-2 \\ \\ 2 \\ 1 \end{array} \begin{array}{c} \begin{array}{ccccccc} N & N-1 & N-2 & & 2 & 1 & 0 \end{array} \\ \left\| \begin{array}{ccccccc} 1-p & p & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1-p & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1-p & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1-p & p & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1-p & p \end{array} \right\| \end{array},$$

…

$$P(m(t)) = \begin{array}{c} \\ N \\ N-1 \\ N-2 \\ \ldots \\ m(t) \end{array} \begin{array}{c} \begin{array}{ccccccc} N & N-1 & \ldots & N-m(t) & \ldots & 1 & 0 \end{array} \\ \left\| \begin{array}{ccccccc} P(0) & P(1) & \ldots & P(m(t)) & \ldots & 0 & 0 \\ 0 & P(0) & \ldots & P(m(t)-1) & \ldots & 0 & 0 \\ 0 & 0 & \ldots & P(m(t)-2) & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & \ldots & P(m(t)-1) & P(m(t)) \end{array} \right\| \end{array}.$$

Структура матриць $P(m(t))$ передбачає виконання умови $N > 2M(t)$. При цьому рядки і стовпці зазначених матриць ідентифікуються відповідним обсягом товару $- N, N-1,\ldots, N-m(t),\ldots,2,1,0$.

Елементи матриці $P(m(t))$ визначаються відповідно до виразу (2.20), тобто

$$P(0) = C_{m(t)}^{0} p^{0} \cdot (1-p)^{m(t)-0} = 1 - p^{m(t)};$$

$$P(1) = C_{m(t)}^{1} p^{1} \cdot (1-p)^{m(t)-1} = m(t) p (1-p)^{m(t)-1};$$

$$P(2) = C_{m(t)}^{2} p^{2} \cdot (1-p)^{m(t)-2} = \frac{m(t)(m(t)-1)}{2!} p^{2} (1-p)^{m(t)-2};$$

$$\ldots$$

$$P(m(t)-1) = C_{m(t)}^{m(t)-1} p^{m(t)-1} \cdot (1-p)^{1} = m(t) p^{m(t)-1} (1-p);$$

$$P(m(t)) = C_{m(t)}^{m(t)} p^{m(t)} \cdot (1-p)^{m(t)-m(t)} = m(t) p^{m(t)}.$$

Очевидно, сума елементів кожного рядка всіх зазначених матриць $\left\| P_{ij}(m(t)) \right\|$ дорівнює одиниці, тобто

$$\sum_{r(t)\in \overline{0,m(t)}} P(r(t)) = \sum_{r(t)\in \overline{0,m(t)}} C_{m(t)}^{r(t)} p^{r(t)} (1-p)^{m(t)-r(t)} = 1,$$

що відповідає умовам визначення автомата з випадковими переходами.

Процедура побудови функції зміни обсягу продажів реалізується на основі двох дій – визначення на основі (2.19) числа покупців $m(t)$, а потім на основі моделі (2.22) стану $z(t)$ за умови, що стан $z(t-1)$ відомий. Ці дії виконуються за допомогою послідовності випадкових рівномірно розподілених чисел в інтервалі (0,1) та методики схеми жереба.

Методикою передбачається вибір деякого числа з послідовності випадкових чисел і порівняння його зі значеннями сум ймовірностей з рядків матриць (2.22), що дозволяє визначити стан $z(t)$, в який перейде модельований об’єкт зі стану $z(t-1)$. Інакше кажучи, дана методика повністю відповідає процесу функціонування автомата з випадковими переходами.

Ординатами функції, що визначається, є значення стану торговельного підприємства в різні моменти часу $z(t)$ $t \in \{1,2,3\ldots\}$. Кожне нове значення $z(t)$ визначається виходячи з відомого попереднього значення $z(t-1)$. При цьому $z(0) = N$ та $z(t-1) \geq z(t)$, що відповідає незростаючому характеру функції, яка визначається.

2.6. Неперервно-стохастичне моделювання

Основою для побудови аналітичних моделей неперервно-стохастичних процесів є теорія систем масового обслуговування (queuing system). Функціонування систем масового обслуговування (СМО) має характер обслуговування заявок, що надходять, або клієнтів. Для виконання дій обслуговування система має необхідне обладнання, яке називають каналами, або лініями обслуговування [21].

Заявки бувають одиничними і такими, що утворюють потік, основною характеристикою якого є закон розподілу надходження заявок у СМО.

Якщо чергова заявка надходить у систему і виявляє всі канали зайнятими, то вона стає в чергу на обслуговування. Основними характеристиками черги є її довжина і порядок вибору заявки з черги.

Процес обслуговування може бути структурований шляхом виділення в ньому таких його частин як стадії. Таким чином, процес обслуговування може бути як одностадійним, так і багатостадійним.

Для реалізації стадії обслуговування виділяється один або декілька каналів обслуговування. В останньому випадку СМО є багатоканальною. У цілому схема багатоканальної, багатостадійної СМО з чергами показана на рис. 2.8.

Вивчення обслуговування окремої заявки полягає у визначенні, чи відбулася подія завершення обслуговування, і у визначенні тривалості обслуговування, а також в оцінені якості обслуговування.

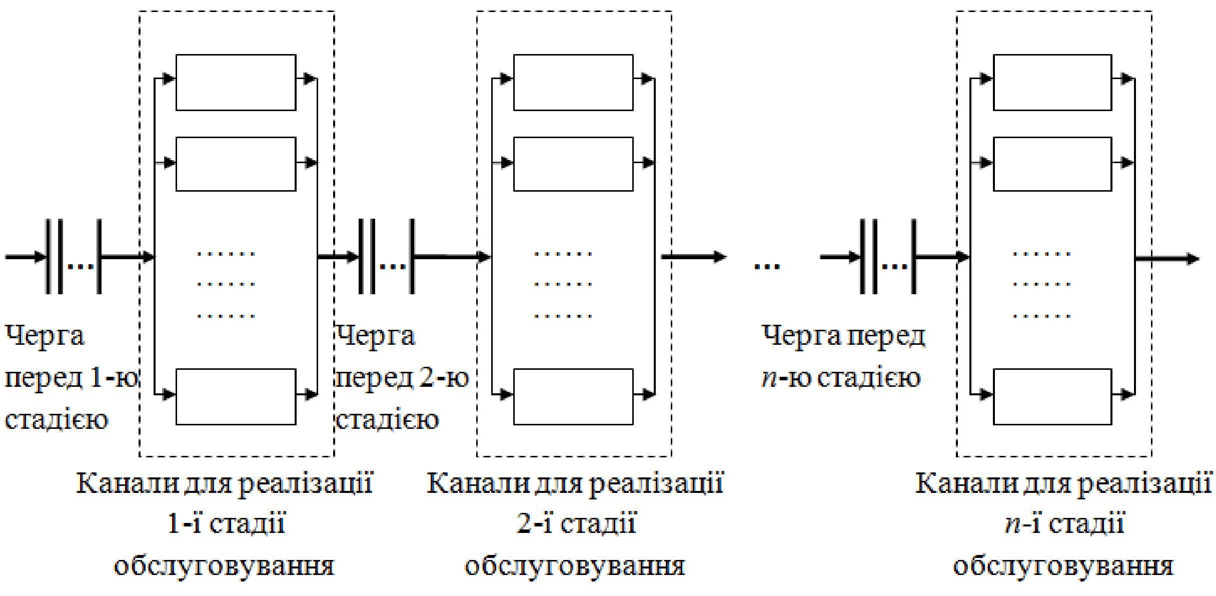


Рисунок 2.8 – Схема багатоканальної, багатостадійної СМО з чергами

При вивченні обслуговування потоку заявок виникають додаткові завдання: визначення частки обслужених заявок і частки заявок, які отримали відмову; визначення часу зайнятості та простою каналів та ін.

Аналітичне моделювання неперервно-стохастичних процесів на основі СМО складається з моделювання потоку заявок і моделювання функціонування обслуговуючих каналів.

Моделювання потоку заявок передбачає визначення або завдання характеру цього потоку. Потік називається потоком однорідних подій, якщо з точки зору обслуговування усі заявки рівноправні і має значення лише факт надходження або відсутності заявки в системі обслуговування в даний момент часу. Кожна заявка в цьому випадку характеризується моментом надходження, а потік заявок – послідовністю моментів $t_1, t_2, ..., t_k$ або законом, що визначає чергування моментів $t_j \left(j \in \overline{1,k}\right)$.

Якщо заявки неоднорідні, то кожна з них, окрім моменту надходження t_j, характеризується набором параметрів $\alpha_{1j}, \alpha_{2j}, ..., \alpha_{mj}$, які виражають властивості заявки. Потік заявок в цьому випадку подається потоком векторів $\left(t_j, \alpha_{1j}, \alpha_{2j}, ..., \alpha_{mj}\right)\left(j \in \overline{1,k}\right)$. Наприклад, для потоку автомобілів, що прибувають на бензозаправку, окрім моменту часу прибуття, важливе значення мають такі властивості автомобілів як ємність бака, вид і марка палива, необхідність надання додаткових послуг. Усі ці властивості виражаються відповідними параметрами. Іншим прикладом може бути потік заготовок, що надходять на оброблення на виробничу ділянку або окремий верстат. У цьому випадку, крім моментів часу надходження окремих заготовок, важливе значення мають ще й такі параметри як вид, клас обробки, число оброблюваних поверхонь, їх складність і т.д.

Для математичного опису випадкового потоку однорідних заявок звичайно використовують закони розподілу не моментів часу t_j, а інтервалів часу між сусідніми моментами – $\xi_j \left(j \in \overline{1,k}\right)$, зв'язок між якими показано на рис. 2.9.

Із зазначеного розподілу t_j, $\xi_j \left(j \in \overline{1,k}\right)$ по осі часу очевидно, що

$$t_1 = 0 + \xi_1,\ t_2 = \xi_1 + \xi_2, ..., t_k = \xi_1 + \xi_2 + ... + \xi_k,$$

тобто завдання закону розподілу інтервалів $\xi_j \left(j \in \overline{1,k}\right)$ дозволяє визначити і значення $t_j \left(j \in \overline{1,k}\right)$.

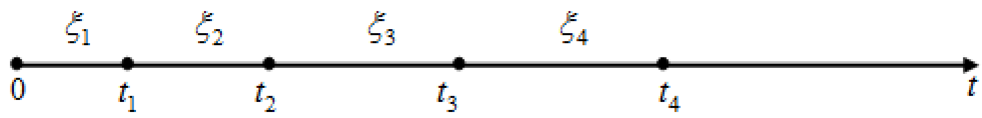


Рисунок 2.9 – Розподіл моментів $t_j\left(j \in \overline{1,k}\right)$ та інтервалів $\xi_j\left(j \in \overline{1,k}\right)$ між ними по осі часу $0t$

Використання закону розподілу інтервалів $\xi_j\left(j \in \overline{1,k}\right)$ замість закону розподілу моментів $t_j\left(j \in \overline{1,k}\right)$ пояснюється неперервністю величин ξ_j і їх простим зв'язком з величинами $t_j\left(j \in \overline{1,k}\right)$. Розглядають інтегральні закони розподілу випадкових величин $\xi_j\left(j \in \overline{1,k}\right)$

$$F(z_1, z_2,..., z_k) = P(\xi_1 < z_1, \xi_2 < z_2,..., \xi_k < z_k)$$

і диференціальні закони розподілу або функції розподілу густини ймовірностей, які зазвичай позначаються як $f(z_1, z_2,..., z_k)$ [21].

Якщо $f(z_1, z_2,..., z_k) = f_1(z_1) \cdot f_2(z_2) \cdot \ldots \cdot f_k(z_k)$, тобто випадкові величини незалежні, то потік називають потоком з обмеженою післядією. Коли, до того ж, закони розподілу $f_j(z_j)$ однакові, то потік називають рекурентним.

Потік називають ординарним, якщо за будь-якого t_0 на осі часу ймовірність появи в інтервалі $(t_0, t_0 + \Delta t)$ двох або більше заявок прагне до нуля при $\Delta t \to 0$.

Якщо ймовірність надходження k заявок в інтервалі $(t_0, t_0 + \Delta t)$ не залежить від t_0, а визначається значеннями k і Δt, то потік називають стаціонарним.

Для рекурентного потоку середнє значення довжини інтервалу між моментами надходження заявок визначається значенням математичного очікування

$$m_z = \int_0^\infty z f(z) dz.$$

Величину $\lambda = 1/m_z$ називають інтенсивністю потоку заявок, яка визначає середнє число заявок, що надходять в одиницю часу.

Якщо потік має властивості стаціонарності, ординарності і відсутності наслідку, то такий потік називають найпростішим або пуасоновським потоком. Ймовірність появи k заявок у пуасоновському потоці в інтервалі Δt визначається виразом

$$P_k(\Delta t) = \frac{(\lambda \Delta t)^k}{k!} e^{-\lambda \Delta t}.$$

Функція густини розподілу ймовірностей в цьому випадку має вигляд

$$f(\Delta t) = \lambda e^{-\lambda \Delta t}.$$

Ця функція має спеціальну назву – показовий закон розподілу.

Таким чином, у цілому моделювання потоку заявок зводиться до завдання або визначення інтегрального або диференціального закону розподілу (функції густини розподілу ймовірностей) інтервалів між моментами надходження заявок.

Інша частина аналітичного моделювання неперервно-стохастичних процесів – моделювання процесу обслуговування полягає у визначенні характеру процесу обслуговування, що більш конкретно означає:

завдання порядку постановки заявки в чергу у випадку зайнятості каналів обслуговування і вибору заявки з черги при звільненні каналу або декількох каналів;

- завдання порядку призначення каналів, що звільнилися, для обслуговування чергової заявки;
- визначення часу обслуговування заявки каналом або декількома каналами.

Відомо, що канал може знаходитися у двох станах – «вільний» і «зайнятий». Заявка, що надійшла в систему обслуговування у момент t_j, приймається до обслуговування, якщо є вільний канал, або ставиться в чергу, якщо немає вільних каналів.

Часто робиться припущення, що заявка може знаходитися в черзі протягом деякого часу Δt. Якщо заявка не буде прийнята до обслуговування до моменту $t_j + \Delta t$, то вона залишає систему. З урахуванням зазначеного припущення розглядають: СМО з відмовами, коли $\Delta t = 0$, СМО з очікуванням, коли $\Delta t = \infty$, і нарешті, змішані СМО, коли Δt – позитивна випадкова величина.

Прикладами порядку постановки заявок у чергу і вибірки їх з неї можуть бути: прийом заявок у порядку надходження, а вибірка в порядку першого з прибулих; використання різних пріоритетів; випадкові порядки.

Порядками призначення звільнених каналів для обслуговування чергової заявки можуть бути: залучення каналів у порядку зростання номера каналу; в порядку звільнення; у випадковому порядку і т.д.

Важливою величиною, що характеризує функціонування каналу, є час обслуговування заявки, чи інакше – час зайнятості каналу обслуговуванням заявки. Це, як правило, випадкова величина з заданим законом розподілу. Наприклад, випадкова величина часу обслуговування τ характеризується експоненціальним законом з густиною розподілу $f(\tau) = \lambda e^{-\lambda\tau}$ $(\tau > 0)$, де λ – параметр, що виражає інтенсивність обслуговування.

Функція розподілу

$$F(\tau) = \int_0^{\tau} \lambda e^{-\lambda t} dt = 1 - e^{-\lambda\tau}$$

являє собою ймовірність того, що обслуговування завершиться за час τ. Або інакше, це ймовірність того, що канал звільниться через час τ.

Оскільки канал може знаходитися в одному з двох станів – «зайнятий» або «вільний», то ймовірність того, що канал буде зайнятий через час τ, визначається як

$$1 - F(\tau) = 1 - 1 + e^{-\lambda\tau} = e^{-\lambda\tau}.$$

Якщо в системі зайняті k каналів, то ймовірність того, що жоден з них не звільниться за час τ, визначається як $\left(e^{-\lambda\tau}\right)^k = e^{-k\lambda\tau}$.

Для СМО з пуасоновським потоком заявок і експоненціальним розподілом часу обслуговування характерна відсутність післядії, тобто майбутній стан залежить тільки від теперішнього і не залежить від того, що відбувалося в минулому. Такі процеси називають марковськими [21]. Вони є теоретичною основою для аналізу великого числа реально існуючих СМО.

2.7. Узагальнені моделі на основі агрегату

Однією з принципових проблем моделювання складних систем є проблема одночасного моделювання різних видів процесів – неперервно-детермінованих, дискретно-детермінованих і т.д. Проблема полягає в тому, що математичні схеми для моделювання різних видів процесів не сполучаються одна з одною, що не дозволяє отримати узагальнену модель складної системи.

Розв'язанням проблеми побудови узагальненої моделі є розроблення універсальної математичної схеми, що дозволяє відображати особливості всіх видів процесів – неперервних, дискретних, детермінованих, стохастичних і різних їх комбінацій. Такою універсальною математичною схемою є агрегативна система або просто агрегат. Цю схему запропонувала група вчених під керівництвом Н.П. Бусленко [14].

Для забезпечення властивості універсальності в агрегаті реалізуються наступні механізми [22]:

- зміни стану динамічної системи під дією внутрішніх причин, що виникають у результаті взаємодії елементів системи між собою;
- прийому вхідного сигналу із зовнішнього середовища і зміни стану під дією цього сигналу;
- формування вихідного сигналу як реакції на зміну стану динамічної системи під впливом внутрішніх причин або сигналу із зовнішнього середовища.

Функціонування динамічної системи і агрегату, що відображає її, зводиться до виконання зазначених механізмів. Для розгляду механізмів припустимо, що в початковий момент часу t_0 динамічна система перебуває у стані z^0, де z^0 – внутрішня точка замкнутої області Z у n-вимірному просторі станів системи.

У рамках механізму зміни стану під дією внутрішніх причин динамічна система переходить зі стану z^0 в інші стани $z_t \in Z$, що відповідають моментам часу $t > t_0$, здійснюючи при цьому рух $z(t)$.

Характер причин, що підтримують рух, не змінюється аж до виходу точки z_t на кордон замкнутої області Z. Нехай моментом часу виходу на кордон буде t^*, а станом $z^* = z_{t^*}$.

Сукупність упорядкованих пар (t, z_t) для $t_0 < t < t^*$, тобто $\{(t, z_t) | t_0 < t < t^*\}$, являє собою фрагмент руху $z(t)$, відповідний інтервалу (t_0, t^*). Для завдання цього фрагмента руху всередині області Z необхідно вказати співвідношення, що визначають значення z_t для $t \in (t_0, t^*)$, при відомих значеннях t_0 і z^0. Наприклад, це може бути диференціальне рівняння

$$\frac{dz}{dt} = f(t, z)$$

з початковими умовами: при $t = t_0$, $z(t_0) = z^0$.

Для визначення t^* і z^* необхідно спільно розв’язати рівняння руху точки z_t і рівняння границі області Z.

Момент t^* називають опорним моментом часу. У цей момент стан системи z^* стрибкоподібно, миттєво змінюється, і система опиняється в стані $z' = z_{t^*+0}$, де $t^* + 0$ – момент часу, близький до t^*.

Вибір стану $z' \in Z$ реалізується випадково, відповідно до розподілу ймовірностей, який залежить від t^* і z^*, що формалізується наступним чином:

$$z' = P\left(z'\left(t^*, z^*\right)\right).$$

Фрагмент руху, пов'язаний з переходом від точки $\left(t^*, z^*\right)$ до точки $\left(t^* + 0, z'\right)$ фазового простору, називають стрибком стану при виході на кордон.

Крім цього, в момент t^* у рамках механізму формування вихідного сигналу визначається вихідний сигнал $y_{t^*} = G^*\left(t^*, z^*\right)$, який залежить від t^* та z^*. Функція G^* є заданою та має іноді випадковий характер. Процес переходу зі стану z^0 в z^* і далі в стан z' показано на рис. 2.10.

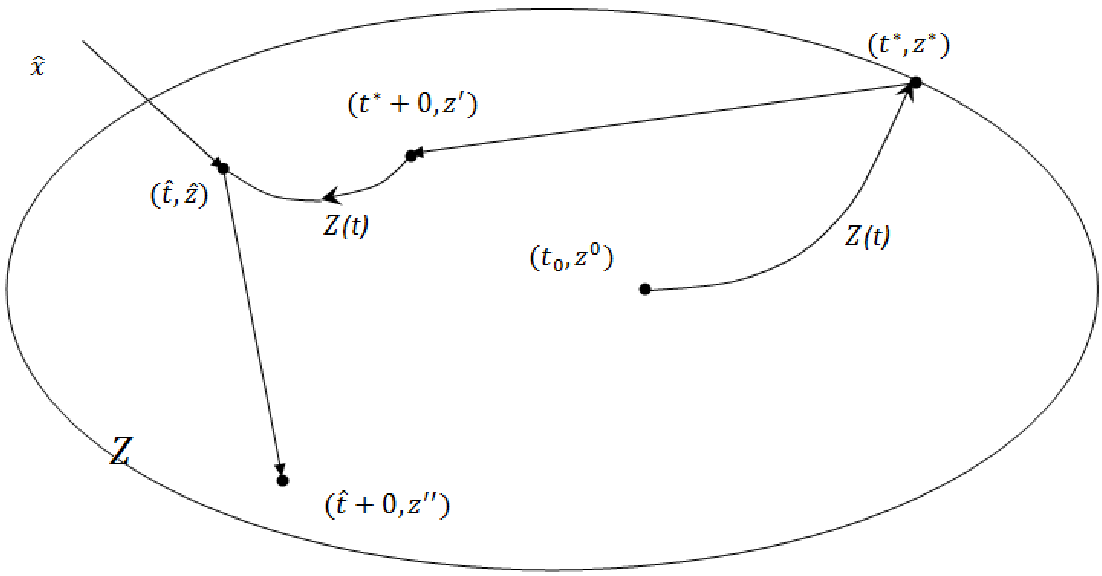


Рисунок 2.10 – Схема зміни станів у рамках механізмів зміни стану під дією внутрішніх причин і під дією вхідного сигналу

Із стану z' під дією знову-таки внутрішніх причин система переходить в інші стани, здійснюючи нове переміщення всередині області Z, новий стрибок стану при виході на кордон і т.д.

Усі переміщення всередині області Z під дією внутрішніх причин припиняються при надходженні вхідного сигналу $\hat{x}$ із зовнішнього середовища. Більш конкретно припустимо, що в момент $\hat{t}$, коли система перебувала у стані $\hat{z}$, надходить вхідний сигнал $\hat{x}$.

У момент $\hat{t}$ припиняється переміщення під дією внутрішніх причин, і зі стану $\hat{z}$ система стрибкоподібно, миттєво переходить у стан $z'' = z_{\hat{t}+0}$, який є внутрішньою точкою області Z.

Вибір стану $z'' \in Z$ здійснюється випадково відповідно до розподілу ймовірностей, який залежить від $\hat{t}, \hat{x}, \hat{z}$, тобто

$$z'' = P\left(z''\left(\hat{t}, \hat{z}, \hat{x}\right)\right).$$

Фрагмент руху, пов'язаний з переходом від точки $\left(\hat{t}, \hat{z}\right)$ до точки $\left(\hat{t} + 0, z''\right)$, називають стрибком стану при надходженні вхідного сигналу.

Момент $\hat{t}$ є також опорним моментом часу. У цей момент крім зазначеного стрибка стану, реалізованого в рамках механізму прийому вхідного сигналу і зміни стану під дією цього сигналу, формується і видається вихідний сигнал

$$y_{\hat{t}} = \hat{G}\left(\hat{t}, \hat{z}, \hat{x}\right),$$

де $\hat{G}$ – заздалегідь відома, можливо, випадкова функція, яка залежить від моменту $\hat{t}$, стану $\hat{z}$ і від вхідного сигналу $\hat{x}$.

Далі зі стану z'' система переходить в інші стани всередині області Z під дією внутрішніх причин, здійснює нові скачки при виході на границю або при надходженні вхідного сигналу із зовнішнього середовища і т.д.

Таким чином, процес функціонування динамічної системи з дискретним втручанням випадку, який відображається за допомогою агрегату, складається з переміщень точки $\left(t, z_t\right)$ всередині області $\left(t, Z\right)$, які можуть перериватися стрибками станів в опорні моменти часу t^* і $\hat{t}$, тобто при виході на границю або при надходженні вхідного сигналу, що і показано на рис. 2.10.

Математично агрегат виражається:

1) рівнянням меж області Z;

2) рівняннями руху точки z_t всередині області Z;

3) співвідношеннями для розрахунку розподілу ймовірностей стрибка стану при виході на границю і під час надходження вхідного сигналу із зовнішнього середовища;

4) співвідношеннями для розрахунку вихідних сигналів при виході на границю і під час надходження вхідного сигналу.

Всі зазначені рівняння і співвідношення є характеристиками загальної динамічної системи з дискретним втручанням випадку. Узагальненість розглянутої динамічної системи виражається в тому, що накладаючи деякі обмеження на зазначені характеристики, можна отримувати раніше розглянуті класи динамічних систем.

Зокрема, наприклад, якщо припустити, що скачки нульові, тобто координати стану перед стрибком і після збігаються, і рух $z(t)$ є необхідну кількість разів диференційованою функцією, то отримуємо клас динамічних систем, процеси в яких мають неперервно-детермінований характер і які описуються відповідною системою звичайних диференціальних рівнянь.

Далі, якщо накласти обмеження на переміщення під дією внутрішніх причин, вважаючи їх нульовими, і всі зміни станів системи звести до стрибків стану під дією вхідних сигналів із зовнішнього середовища, то отримуємо клас динамічних систем, процеси в яких мають дискретно-стохастичний характер і які описуються на основі імовірнісного автомата.

Якщо до зазначених вище обмежень додати обмеження, які полягають в тому, що розподіл ймовірностей стрибка станів являє собою набір нулів і одиниць, то отримуємо клас динамічних систем, процеси в яких мають дискретно-детермінований характер і які описуються на основі кінцевих автоматів.

Найбільш близькими до загальної динамічної системи з дискретним втручанням випадку, описуваної за допомогою агрегату, є динамічні системи, які описуються на основі теорії систем масового обслуговування. Для цих систем характерним є ненульові переміщення всередині області Z під дією внутрішніх причин і ненульові стрибки станів.

Характеристики загальної динамічної системи з дискретним втручанням випадку, тобто відповідні рівняння і співвідношення, вибираються із заданих класів аналітичних виразів. Наприклад, рівняння можуть бути обрані лінійними або нелінійними, з постійними або змінними коефіцієнтами і т.д. Співвідношення для розрахунку розподілу ймовірностей стрибка стану можуть враховувати і не враховувати передісторію процесу.

Конкретизація характеристик, тобто їх вибір із заданих класів, дозволяє виділити деякі найбільш часто використовувані види агрегатів для опису загальної динамічної системи з дискретним втручанням випадку. До таких найбільш часто використовуваних видів агрегатів належить кусково-лінійний агрегат (КЛА).

Для завдання КЛА необхідні всі характеристики загальної динамічної системи з дискретним втручанням випадку. Конкретизація характеристик для КЛА полягає в тому, що множина станів Z розділяється на частини або куски у вигляді непересічних підмножин z_k. Границі z_k визначаються лінійними рівняннями, які задають в n-вимірному евклідовому просторі гіперплощини, які є сторонами багатогранника. Таким чином, підмножинами z_k є багатогранники в евклідовому просторі. У двовимірному евклідовому просторі підмножинами z_k є багатокутники, приблизне їх розташування показано на рис. 2.11. Основними координатами евклідового простору на рис. 2.11 є z_{01}, z_{02}. Неперетинні багатокутники позначаються $z_1, z_2,..., z_\upsilon,..., z_k$. Кожен багатокутник $z_\upsilon\left(\upsilon \in \overline{1,k}\right)$ має місцеву систему координат – $z_{\upsilon 1}, z_{\upsilon 2}$.

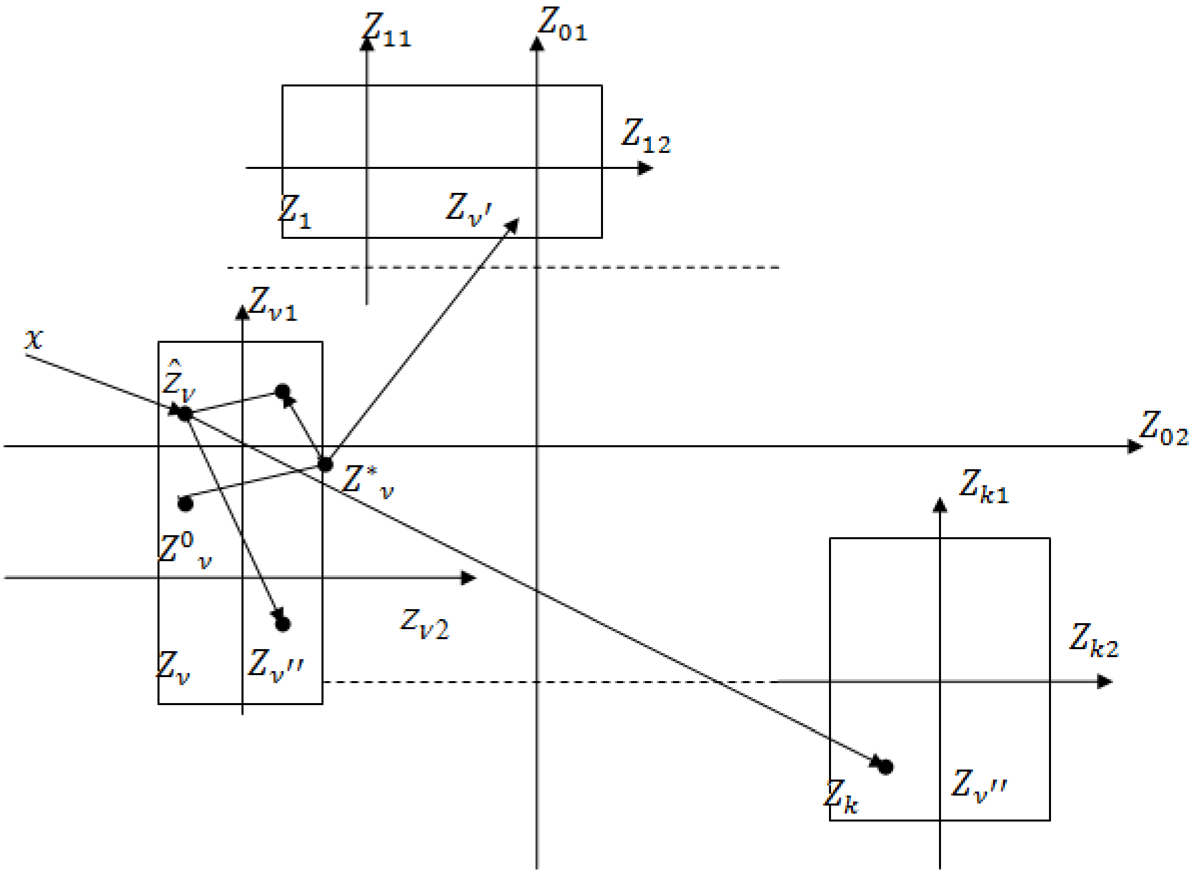


Рисунок 2.11 – Розподіл підмножин станів КЛА у 2-вимірному евклідовому просторі

Стан КЛА подається у вигляді $z = (\upsilon, z_\upsilon)$, де υ – індекс підмножини (багатогранника), а z_υ – точка всередині багатогранника з координатами в місцевій системі координат – $z_{\upsilon 1}, z_{\upsilon 2},..., z_{\upsilon n}$. У зображеному на рис. 2.11 двовимірному випадку місцева система має дві координати – $z_{\upsilon 1}, z_{\upsilon 2}$, тобто стан КЛА в цьому випадку визначається як $z = (\upsilon, z_{\upsilon 1}, z_{\upsilon 2})$.

Компоненту υ називають дискретною складовою стану або основним станом, а вектор z_υ – вектором додаткових координат.

Подальша конкретизація характеристик КЛА зв’язується з процесом функціонування – із зазначенням конкретних співвідношень та рівнянь, що описують переміщення і стрибки всередині багатогранників.

Нехай у початковий момент t_0 КЛА перебуває в стані $z^0 = \left(\upsilon, z^0{}_\upsilon\right)$. Можливе розташування точки $z^0{}_\upsilon$ для двовимірного випадку показано на рис. 2.11. При $t > t_0$ здійснюється рух $z^0{}_\upsilon(t)$ всередині багатогранника υ. У разі КЛА рух задається лінійними рівняннями вигляду

$$\frac{dz_\upsilon}{dt} = \alpha_\upsilon,$$

де $\alpha_\upsilon = (\alpha_{\upsilon 1}, \alpha_{\upsilon 2} .. \alpha_{\upsilon n})$ – постійний вектор.

Розв’язання зазначеного рівняння має вигляді

$$z_\upsilon(t) = z^0{}_\upsilon + \alpha_\upsilon (t - t^0),$$

тобто переміщення точки всередині багатогранника реалізується за лінійним законом для двовимірного випадку (рис. 2.11).

Рух $z_\upsilon^0(t)$ всередині багатогранника за лінійним законом продовжується до тих пір, поки траєкторія руху не досягне границі багатогранника в точці z_υ^*, що і показано на рис. 2.11.

Точка z_υ^* досягається в момент t^*, який, як і раніше, визначається опорним. У цей момент часу відбуваються дві важливі дії: миттєвий стрибок стану в результаті виходу на кордон і видача вихідного сигналу.

Миттєвий стрибок стану відбувається з точки (υ, z_υ^*) в точку $(\upsilon', z_{\upsilon'})$, яка є внутрішньою точкою того самого або нового багатогранника. Миттєвий стрибок показується на рис. 2.11 у вигляді стрілок, направлених в той же або в перший багатогранник, тобто $\upsilon' = \upsilon$ або $\upsilon' = 1$.

Нова дискретна складова υ' вибирається випадковим чином відповідно до розподілу ймовірностей $P_{\upsilon\upsilon'}$, який може залежати, наприклад, від індексу багатогранника υ , від точки виходу на границю z_υ^*, від номера грані, на якій розташовується координата z_υ^* і т.д.

З точки $(\upsilon', z_{\upsilon'})$ здійснюється рух $z_{\upsilon'}(t)$, як і раніше з точки $(\upsilon, z^0{}_\upsilon)$.

Як йшлося вище, другою дією при виході на границю є видача вихідного сигналу.

Вихідний сигнал $y = (\alpha, y_\alpha)$ має дискретну складову α, яка зазвичай визначає вид сигналу – звуковий, світловий та ін., і вектор додаткових координат $y_\alpha = (y_{\alpha 1}, y_{\alpha 2}, .. y_{\alpha n})$, компоненти якого є характеристиками сигналу.

Рух всередині багатогранника може перериватися в результаті надходження вхідного сигналу із зовнішнього середовища.

Нехай в момент $\hat{t}$, який є опорним, із зовнішнього середовища надходить вхідний сигнал $x = (\mu, x_\mu)$, де μ – дискретна складова сигналу, що визначає, як правило, вид сигналу, і $x_\mu = (x_{\mu 1}, x_{\mu 2} ... x_{\mu n})$ – вектор додаткових координат, компоненти якого є характеристиками сигналу, виражаючи, наприклад, частоту, амплітуду, потужність і т.п. сигналу.

В момент $\hat{t}$, коли агрегат перебував у стані $(\upsilon, \hat{z}_\upsilon)$, де $\hat{z}_\upsilon = \hat{z}_\upsilon(\hat{t})$, рух усередині багатогранника припиняється, і знову відбуваються дві дії: миттєвий стрибок стану в результаті надходження вхідного сигналу і видача вихідного сигналу.

Миттєвий стрибок стану відбувається з точки $(\upsilon, \hat{z}_\upsilon)$ в точку $(\upsilon'', \hat{z}_{\upsilon''})$, яка може бути внутрішньою точкою того самого або іншого багатогранника. Надходження вхідного сигналу x і викликаний ним миттєвий стрибок показано на рис. 2.11 у вигляді стрілки x і стрілок, що виходять з $\hat{z}_\upsilon$ і спрямовані у той же самий багатогранник $\upsilon'' = \upsilon$ або в інший багатогранник, наприклад $\upsilon'' = k$.

Вибір дискретної складової υ'' реалізується випадковим чином на основі розподілу ймовірностей $Q_{\upsilon\upsilon''}$, який може залежати від $\upsilon, \hat{z}_\upsilon, \mu$.

Вихідний сигнал формується, як і у випадку виходу точки на границю багатогранника, тобто у вигляді $y = (\alpha, y_\alpha)$.

Далі відбувається переміщення $z_{\upsilon''}(t)$ всередині багатогранника $z_{\upsilon''}$, яке переходить у новий стрибок стану через досягнення межі багатогранника або через надходження вхідного сигналу із зовнішнього середовища.

2.8. Модель сполучення елементів у складній системі

Будь-яка складна система, як правило, передбачає наявність великої кількості елементів, між якими встановлюються різноманітні зв'язки.

По зв'язках у вигляді деяких сигналів передається інформація, яка використовується для функціонування як окремих елементів, так і для системи в цілому.

Зв'язки забезпечують «міцність» системи і тому є важливим «будівельним матеріалом» для системи, який обов'язково повинен враховуватися при побудові моделі системи. Наявність зв'язків якраз і враховується в моделях сполучення.

При побудові моделей сполучення використовуються наступні передумови і припущення [22].

Щоб зосередитися на зв'язках між елементами, самі елементи подаються у вигляді моделі «чорний ящик» – на вхід елемента подається вхідний сигнал $x = (x_1, x_2..x_n)$, де кожна характеристика вхідного сигналу

$x_i(i \in \overline{1,n})$ має свою область визначення X_i, тобто $x_i \in X_i$, а на виході елемента виникає вихідний сигнал $y = (y_1, y_2,..., y_r)$, де $y_l \in Y_l$ $(l \in \overline{1,r})$, тобто кожна характеристика вихідного сигналу також має свою область визначення.

Для прийому вхідного сигналу $x = (x_1, x_2..x_n)$ елемент системи $e_j(j \in \overline{1,N})$ має набір вхідних контактів $\left[X_i^j\right]_1^{n_j}$, де контакт X_i^j призначений для прийому *i*-ї характеристики вихідного сигналу, тобто для прийому x_i, а n_j – число вхідних контактів, яке у кожного елемента може бути індивідуальним.

Для видачі вихідного сигналу $y = (y_1, y_2..y_r)$ елемент системи $e_j(j \in \overline{1,N})$ має набір вихідних контактів $\left[Y_l^j\right]_1^{r_j}$, де контакт Y_l^j призначається для видачі *l*-ї характеристики вихідного сигналу, тобто для видачі $y_l(l \in \overline{1,r})$, а r_j – число вихідних контактів, яке у кожного елемента може бути індивідуальним.

Таким чином, елемент системи подається схемою на рис. 2.12.

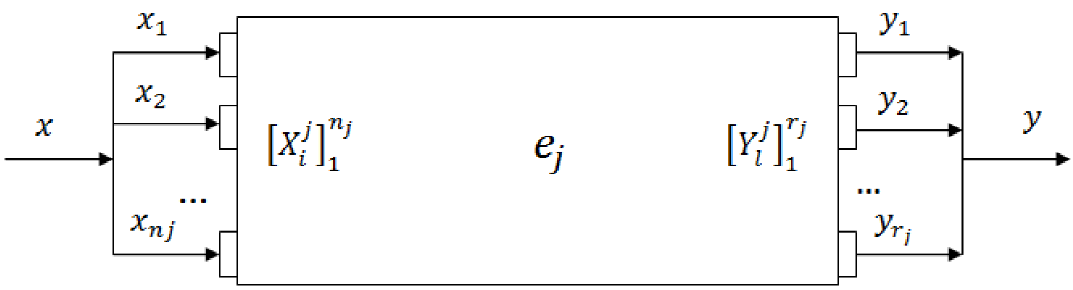


Рисунок 2.12 – Графічне подання елемента системи в моделях сполучення

Припускається далі, що сигнали передаються по зв'язках у вигляді елементарних каналів незалежно один від одного. При цьому до кожного вхідного контакту підключається не більше одного каналу, а з вихідного контакту виходить будь-яке число каналів за умови, що вони направляються до різних елементів системи. Зроблене припущення дозволяє уникнути накладення сигналів, що надходять від різних джерел, один на одний.

Зовнішнє середовище для розглянутої системи у вигляді сукупності елементів $S = \{e_j \mid j \in \overline{1,N}\}$ може бути подане двома основними способами.

Відповідно до першого способу зовнішнє середовище уявляється одним окремо обраним елементом. Зазвичай для цього елемента $j = 0$, тобто зовнішнє середовище визначене елементом e_0, який, як і елементи системи, характеризується наборами вхідних і вихідних контактів – $[X_i^o]_1^{n_0}$, $[Y_l^0]_1^{r_0}$.

Якщо зовнішнє середовище має вигляд окремо обраного елемента, то вся множина вхідних контактів і множина вихідних контактів усіх елементів системи і зовнішнього середовища визначаються як

$$[X_i^j]_N = \bigcup_{j \in \overline{0,N}} [X_i^j]_1^{n_j}; \qquad [Y_l^j]_N = \bigcup_{j \in \overline{0,N}} [Y_l^j]_1^{r_j}.$$

Відповідно до другого способу зовнішнє середовище подається множиною елементів, що називаються джерелами сигналів, і множиною елементів, що називаються споживачами (приймачами) сигналів. Кожне джерело сигналів $\{\hat{e}_{0k} \mid k \in \overline{1,k^*}\}$ характеризується набором вихідних контактів $[Y_l^{Ok}]_i^{r_k} (k \in \overline{1,k^*})$, а кожен споживач сигналів $\{\tilde{e}_{oj} \mid j \in \overline{1,j^*}\}$ характеризується набором вхідних контактів $[X_i^{oj}]_i^{n_j} (j \in \overline{1,j^*})$.

Якщо зовнішнє середовище подається множинами джерел і споживачів, то в цьому випадку множини вхідних і вихідних контактів усіх елементів системи і зовнішнього середовища визначаються як

$$[X_i^j]_N = \{\bigcup_{j \in \overline{1,N}} [X_i^j]_1^{n_j}\} \cup \{\bigcup_{j \in \overline{1,j^*}} [X_i^{oj}]_1^{n_j}\};$$

$$[Y_l^j]_N = \{\bigcup_{j \in \overline{1,N}} [Y_l^j]_1^{r_j}\} \cup \{\bigcup_{k \in \overline{1,k^*}} [Y_l^{ok}]_1^{r_j}\}.$$

На основі множин $[X_i^j]_N$ та $[Y_l^j]_N$ вводиться оператор *R*, що реалізує відображення $R: [X_i^j]_N \to [Y_l^j]_N, (Y_l^k = R(X_i^j))$. Оператор *R* кожному вхідному контакту X_i^j ставить у відповідність вихідний контакт Y_l^k. Фізично це означає, що контакти X_i^j, Y_l^k зв'язуються елементарним каналом.

Наприклад, нехай є система $S = \{e_j \mid j \in \overline{1,5}\}$. Зовнішнє середовище включає два джерела $\{\hat{e}_{Ok} \mid K \in \overline{1,2}\}$ і три споживача інформації $\{\tilde{e}_{Oj} \mid j \in \overline{1,3}\}$. Між елементами системи, джерелами і споживачами зовнішнього середовища є зв'язки, які показані на рис. 2.13.

Найбільш часто використовується завдання оператора сполучення *R* у формі таблиці, рядки якого відповідають номерам вхідних контактів (i), а стовпці відповідають номерам елементів системи і зовнішнього середовища з вхідними контактами (j).

На перетині рядка з індексом *i* та стовпця з індексом *j*, що відповідає вхідному контакту X_i^j, записується пара індексів *(k,l)*, де *k* вказує на номер елемента системи або зовнішнього середовища, а *l* – на номер вихідного контакту Y_l^k, з яким контакт X_i^j зв'язується елементарним каналом.

Таблиця, що задає оператор сполучення *R* для розглянутого прикладу системи і зовнішнього середовища, показаних на рис. 2.13, має наступний вигляд (табл. 2.7).

Таблиця 2.7 – Таблична форма завдання оператора R

i	j							
	1	2	3	4	5	01	02	03
1	01,1	02,1	1,1	1,3	1,3	3,1	5,1	4,2
2	01,2	02,2	1,2	2,1	4,1	3,2		
3		02,3	1,3					

Процес заповнення табл. 2.7 полягає у визначенні пари індексів *(k,l)* на перетині рядків з індексом *i* і стовпців з індексом *j*. Наприклад, нехай *i* = 1, *j* = 1, що означає фіксацію першого входу (*i* = 1) у першого елемента $e_1\,(j=1) - X_1^1$. Згідно зі схемою на рис. 2.13, вхід X_1^1 з'єднується елементарним каналом з першим вихідним контактом (*l* = 1) елемента зовнішнього середовища $\hat{e}_{01}$, тобто *k* = 01. Таким чином, у клітці на перетині першого рядка (*i* = 1) і першого стовпця (*j* = 1) необхідно записати пару (*k*, *l*) = (01,1).

Іншою формою завдання оператора сполучення *R* може бути матриця інцидентності. Таке уявлення є більш громіздким, ніж у формі таблиці, але воно дозволяє використовувати добре розроблені і відомі методи теорії графів для аналізу структури зв'язків.

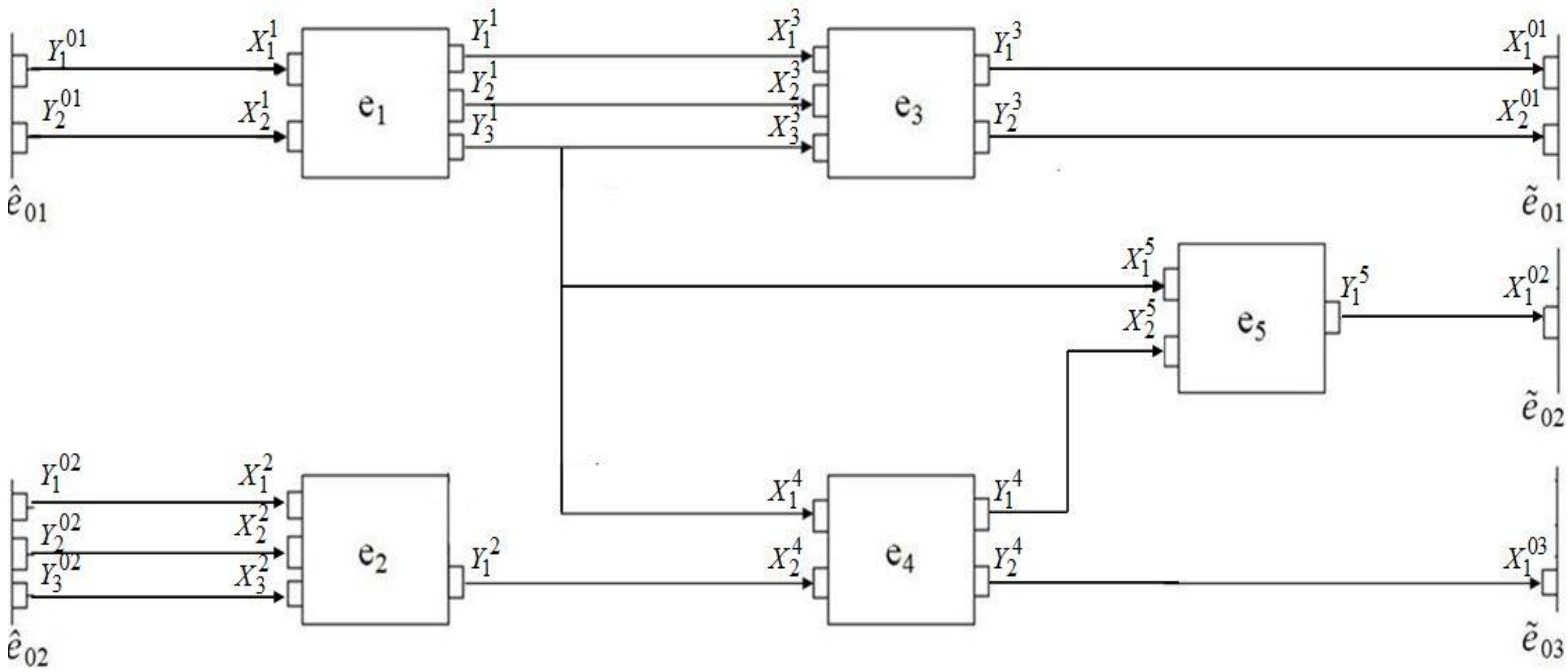


Рисунок 2.13 – Приклад зв’язків між елементами системи і зовнішнім середовищем

Сукупність множин $[X_i^j]_1^{n_j}$, $[Y_l^j]_1^{r_j}$ для всіх елементів системи $S = \left\{\left\langle e_j \,\middle|\, j \in \overline{1,N}\right\rangle\right\}$ і зовнішнього середовища $\left\{\left\langle \hat{e}_{ok} \,\middle|\, k \in \overline{1,k^*}\right\rangle\right\}, \left\{\left\langle \tilde{e}_{oj} \,\middle|\, j \in \overline{1,j^*}\right\rangle\right\}$ разом з оператором сполучення R складають схему сполучення в канонічній формі. Така схема сполучення в канонічній формі є основною моделлю сполучення для завдання зв'язків між елементами системи і зовнішнього середовища.

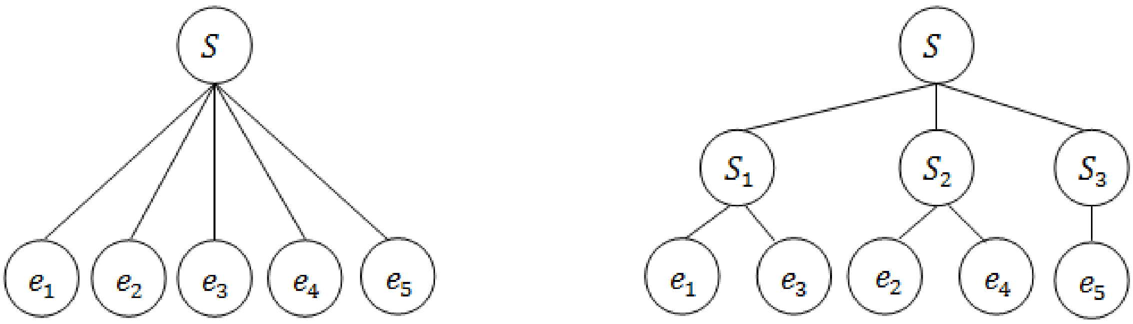
Моделі сполучення бувають одно- і багаторівневими, залежно від того, чи входять елементи системи безпосередньо в систему або до підсистеми, які розподіляються за рівнями ієрархії.

Модель сполучення елементів системи на рис. 2.13 є, очевидно, однорівневою, оскільки елементи системи входять у систему безпосередньо.

Для побудови багаторівневої моделі сполучення необхідно розподілити елементи за підсистемами. Нехай для визначеності елементи $e_j(j \in \overline{1,N})$ розподіляються по підсистемах $S_\mu(\mu \in \overline{1,M})$ так, що кожний елемент e_j входить тільки в одну з підсистем S_μ, які містять не менше одного елемента.

Наприклад, у системі S на рис. 2.13 можливий наступний розподіл елементів за підсистемами: $S_1 = \{e_1, e_3\}, S_2 = \{e_2, e_4\}, S_3 = \{e_5\}$, тобто система уявляється як $S = \left\{\left\langle S_\mu \,\middle|\, \mu \in \overline{1,3}\right\rangle\right\}$.

Різниця між структурами системи, коли елементи безпосередньо входять до неї і коли елементи утворюють підсистеми, показана на рис. 2.14.



а *б*

Рисунок 2.14 – Варіанти структур системи, коли елементи безпосередньо входять до неї (*а*) і коли елементи об'єднуються в підсистеми (*б*)

Багаторівнева модель сполучення, коли елементи системи об'єднуються певним чином у підсистеми, створюється з обов'язковим урахуванням принципу подвійності. Згідно з цим кожна підсистема розглядається з двох точок зору: з одного боку – як система, а з іншого – як елемент.

Підсистема $S_\mu(\mu \in \overline{1,M})$ як система має певну структуру, тобто елементний склад зі зв'язками між елементами і зв'язками із зовнішнім середовищем. При цьому зв'язки із зовнішнім середовищем реалізуються через контакти $X_i^{o\mu}$ і $Y_l^{o\mu}$, що характеризують відповідно споживачів і джерела зовнішнього середовища.

Підсистема $S_\mu(\mu \in \overline{1,M})$ як елемент уявляється моделлю типу «чорний ящик», тобто характеризується лише вхідними $\hat{X}_i^\mu$ і вихідними $\hat{Y}_l^\mu$ контактами для зв'язку з іншими підсистемами і зовнішнім середовищем.

Таким чином, розгляд підсистеми $S_\mu(\mu \in \overline{1,M})$ з урахуванням принципу подвійності призводить до необхідності введення «подвійних» контактів на межі підсистеми.

Наприклад, у системі *S*, в якій виділено три підсистеми S_1, S_2, S_3, межі підсистем з «подвійними» контактами показано на рис. 2.15.

При цьому точками виникнення «подвійних» контактів є точки перетину контуру межі (пунктирна лінія) зі зв'язками даної підсистеми з іншими підсистемами та зовнішнім середовищем.

Візьмемо підсистему $S_1 = \{e_1, e_3\}$, і будемо спочатку розглядати її як систему. Елементи e_1, e_3, що входять в S_1, мають зв'язки між собою і з зовнішнім середовищем, показаним, як звичайно, джерелами і споживачами сигналів. Вихідні контакти Y_1^{01}, Y_2^{01} відносяться до джерела, а вхідні $X_1^{01}, X_2^{01}, X_3^{01}$ – до споживача. При цьому зовнішнє середовище для підсистеми S_1 як системи утворюють інші підсистеми S_2 і S_3, а також раніше визначене зовнішнє середовище у вигляді джерел $\hat{e}_{01}, \hat{e}_{02}$ і споживачів $\tilde{e}_{01}, \tilde{e}_{02}, \tilde{e}_{03}$. Саме від такого зовнішнього середовища підсистема S_1 як система отримує сигнали через контакти Y_1^{01}, Y_2^{01} і видає сигнали через контакти $X_1^{01}, X_2^{01}, X_3^{01}$.

Тепер розглянемо підсистему S_1 як елемент, тобто як модель типу «чорний ящик». У цьому випадку нас не цікавить структура S_1, тобто її склад і зв'язки між її елементами, а цікавими є лише вхідні та вихідні контакти. Вхідними контактами є $\hat{X}_1^1, \hat{X}_2^1$, а вихідними – $\hat{Y}_1^1, \hat{Y}_2^1, \hat{Y}_3^1$.

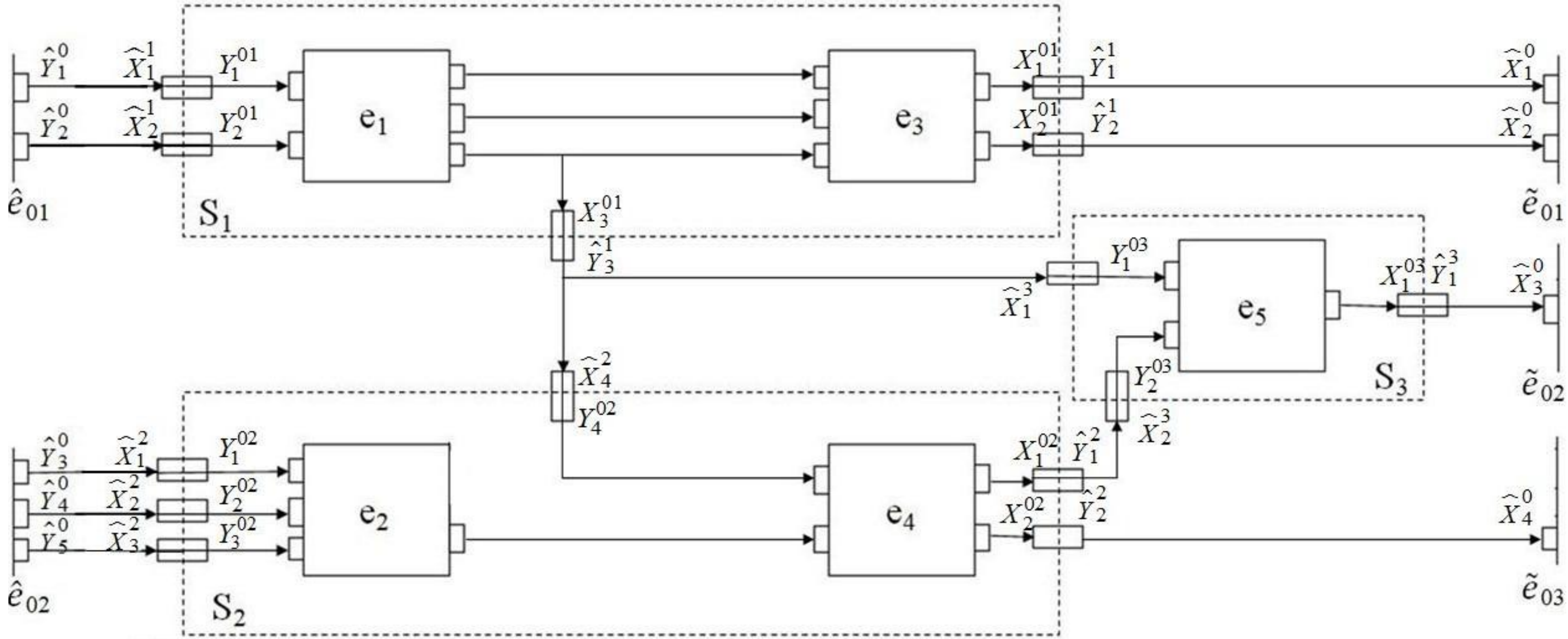


Рисунок 2.15 – Структура системи з підсистемами, на межі яких виникають «подвійні» контакти

Зіставляючи контакти підсистеми S_1 як елемента і раніше виділені контакти підсистеми S_1 як системи, отримуємо в точках перетину межі підсистеми S_1 зі зв'язками з іншими підсистемами і зовнішнім середовищем пари контактів:

$$\left(\hat{X}_1^1, Y_1^{01}\right), \left(\hat{X}_2^1, Y_2^{01}\right), \left(X_1^{01}, \hat{Y}_1^1\right), \left(X_2^{01}, \hat{Y}_2^1\right), \left(X_3^{01}, \hat{Y}_3^1\right).$$

Ці пари контактів і є «подвійними» контактами підсистеми S_1.

На основі аналогічних міркувань визначаються «подвійні» контакти і для інших підсистем S_2, S_3 розглянутого прикладу системи S:

$$\left(\hat{X}_1^2, Y_1^{02}\right), \left(\hat{X}_2^2, Y_2^{02}\right), \left(\hat{X}_3^2, Y_3^{02}\right), \left(\hat{X}_4^2, Y_4^{02}\right), \left(X_1^{02}, \hat{Y}_1^2\right), \left(X_2^{02}, \hat{Y}_2^2\right), \left(\hat{X}_1^3, Y_1^{03}\right), \left(\hat{X}_2^3, Y_2^{03}\right), \left(X_1^{03}, \hat{Y}_1^3\right).$$

Отримані «подвійні» контакти включаються до відповідних наборів контактів. Так контакти виду $X_i^{0\mu}$ – в набори $\left[X_i^{0\mu}\right]$, виду $\hat{Y}_l^{\mu}$ – в набори $\left[\hat{Y}_l^{\mu}\right]$, виду $Y_l^{0\mu}$ – в набори $\left[Y_l^{0\mu}\right]$ і, нарешті, виду $\hat{X}_i^{\mu}$ – в набори $\left[\hat{X}_i^{\mu}\right]$.

Розглянемо тепер підсистеми S_μ як систему і як елемент з урахуванням введених «подвійних» контактів або з урахуванням отриманих наборів $\left[X_i^{0\mu}\right], \left[Y_l^{0\mu}\right], \left[\hat{X}_i^{\mu}\right], \left[\hat{Y}_l^{\mu}\right]$. Елементи підсистеми S_μ як системи, як і раніше, характеризуються наборами вхідних $\left[X_i^j\right]_i^{n_j}$ і вихідних $\left[Y_l^j\right]_1^{r_j}$ контактів усіх тих елементів, які утворюють S_μ, тобто всіх $e_j \in S_\mu$. Зовнішнє середовище для підсистеми S_μ містить набори контактів $\left[X_i^{0\mu}\right], \left[Y_l^{0\mu}\right]$.

Уводиться оператор R_μ, такий, що $Y_l^k = R_\mu\left(X_i^j\right)$. Областю визначення оператора R_μ є множина $\left[X_i^{0\mu}\right] \cup \left\{ \bigcup\limits_{e_j \in S\mu} \left[X_i^j\right]_1^{n_j} \right\}$, а областю значень є множина контактів $\left[Y_l^{0\mu}\right] \cup \left\{ \bigcup\limits_{e_k \in S\mu} \left[Y_l^k\right]_i^{r_k} \right\}$.

Оператор R_μ називається внутрішнім оператором сполучення підсистеми S_μ. Він описує зв'язки між елементами, що утворюють підсистему S_μ, і зовнішнім середовищем, що подається наборами контактів $\left[X_i^{0\mu}\right], \left[Y_l^{0\mu}\right]$.

Оператор R_μ задається найчастіше в табличній формі, аналогічній таблиці для оператора сполучення R однорівневої моделі сполучення (табл. 2.7).

Таблиці, що задають оператори R_μ підсистем S_μ розглянутого прикладу (рис. 2.15), мають наступний вигляд (табл. 2.8 – 2.10).

Таблиця 2.8 – Оператор сполучення R_1

i	j		
	0	1	3
1	3,1	01,1	1,1
2	3,2	01,2	1,2
3	1,3		1,3

Таблиця 2.9 – Оператор сполучення R_2

i	j		
	0	2	4
1	4,1	02,1	02,4
2	4,2	02,2	2,1
3		02,3	

Таблиця 2.10 – Оператор сполучення R_3

i	j		
	0	5	
1	5,1	03,1	
2		03,2	

Підсистема S_μ як елемент системи S характеризується наборами вхідних $\left[\hat{X}_i^\mu\right]$ та вихідних $\left[\hat{Y}_l^\mu\right]$ контактів. Зовнішнім середовищем для підсистеми S_μ у цьому випадку є елементи-джерела $\hat{e}_{01}, \hat{e}_{02}$ і елементи-споживачі $\tilde{e}_{01}, \tilde{e}_{02}, \tilde{e}_{03}$, які характеризуються наборами контактів $\left[\hat{X}_i^0\right], \left[\hat{Y}_l^0\right]$.

Вводиться оператор R^2, такий, що $\hat{Y}_l^\upsilon = R^2\left(\hat{X}_i^\mu\right)$, областю визначення оператора R^2 є множина $\bigcup\limits_{\mu\in\overline{0,M}}\left[\hat{X}_i^\mu\right]$, а областю значень – множина $\bigcup\limits_{\upsilon\in\overline{0,M}}\left[\hat{Y}_l^\upsilon\right]$.

Оператор R^2 називається оператором сполучення підсистем у системі S і задається, як і раніше розглянуті оператори, в табличній формі. Для розглянутого прикладу системи на рис. 2.15 це показано в табл. 2.11.

Таблиця 2.11 – Оператор сполучення R^2

i	μ			
	0	1	2	3
1	1,1	0,1	0,3	1,3
2	1,2	0,2	0,4	2,1
3	3,1		0,5	
4	2,2		1,3	

Набори контактів $\left[\hat{X}_i^{\mu}\right],\left[\hat{Y}_l^{\mu}\right]$ та оператор сполучення R^2 утворюють модель сполучення другого рівня для системи *S*.

Сукупність внутрішніх однорівневих моделей сполучення всіх підсистем $S_{\mu}\left(\mu\in\overline{1,M}\right)$ разом з моделлю сполучення другого рівня утворюють дворівневу модель сполучення системи *S*, яка повністю характеризує структуру зв'язків між елементами системи.

Підсистеми S_{μ} можуть об'єднуватися в більші підсистеми, що приводить до необхідності отримання трирівневої моделі сполучення з оператором сполучення R^3 на третьому рівні. Продовжуючи об'єднувати аналогічним чином, отримуємо багаторівневі моделі сполучення елементів у складних системах.

Складність систем іноді виражається у змінності, керованості і стохастичності структури.

Зазначені властивості структури складної системи необхідно враховувати, наприклад, у таких випадках:

- у ході військової операції, коли відбувається перекидання або перепідпорядкування військових підрозділів та засобів їх забезпечення залежно від ситуації;
- у процесі функціонування гнучких виробничих систем, призначених для випуску декількох видів продукції, при переході з випуску одного виду продукції на інший;
- у процесі функціонування багатопроцесорних комп'ютерів, операційна система яких забезпечує перерозподіл навантаження обчислювальних ресурсів залежно від характеру розв'язуваної задачі.

Під змінністю структури складної системи розуміється змінність у часі наборів вхідних і вихідних контактів як окремих елементів, так і підсистем у багаторівневих системах, а також змінність у часі всіх видів операторів сполучення *R*.

Облік змін у часі наборів вихідних і вхідних контактів полягає в тому, що з плином часу одні контакти зникають, а інші можуть з'являтися.

Залежність оператора сполучення *R* від часу забезпечується множиною таблиць, кожна з яких дійсна на деякому інтервалі $\left(t_i,t_{i+1}\right)$ і змінюється при переході до наступного інтервалу.

Під керованістю структури складної системи розуміється зміна наборів вхідних і вихідних контактів елементів і підсистем, а також залежність операторів сполучень від деякого параметра $\alpha\in A$, де A – множина значень α. Наприклад, при побудові моделі гнучкої виробничої дільниці як параметр управління може розглядатися інтенсивність надходження заготовок.

У тому випадку, коли α є деякою випадковою подією, а A – множина подій, на якій задається розподіл ймовірностей появи подій, то можна говорити про стохастичність структури складної системи. Елементний склад і зв'язки між елементами такої системи мають випадковий характер. Розглядаються задачі, пов'язані з оцінкою впливу законів розподілу на характер функціонування систем.

Моделі сполучення у вигляді схеми сполучення в канонічній формі використовуються для розв'язання будь-яких задач аналізу структури сполучення елементів складних систем. Однак для розв'язання деяких завдань доцільно скористатися моделлю сполучення у вигляді так званої схеми сполучення в натуральній формі. Доцільність пояснюється спрощенням розв'язання задач аналізу.

Основним поняттям у моделях сполучення у вигляді схем сполучення в натуральній формі є поняття штекера як сукупності контактів елемента e_j, пов'язаних елементарними каналами з контактами елемента e_k.

Нехай маємо систему $S = \{e_j \mid j \in \overline{1,N}\}$, зовнішнім середовищем для якої є K^* зовнішніх джерел з набором вихідних контактів $\left[Y_l^{0k}\right]_1^{2k}$ $\left(k \in \overline{1,K^*}\right)$ і j^* споживачів інформації з набором вхідних контактів $\left[X_i^{0j}\right]_1^{n_j}\left(j \in \overline{1,j^*}\right)$.

Розглянемо спочатку особливості побудови натуральної форми однорівневої схеми сполучення, тобто випадок, коли елементи системи не розподіляються за підсистемами, а безпосередньо входять у систему.

Візьмемо елемент e_j $\left(j \in \overline{1,N}\right)$ або $\left(j \in \overline{01,0j^*}\right)$, який належить до системи або до зовнішніх споживачів, і елемент e_k, який належить до системи або до зовнішніх джерел. Визначимо множину вхідних контактів $\left[X^{jk}\right]$ елемента e_j, пов'язаних з елементом e_k. Якщо множина $\left[X^{jk}\right] \neq 0$, тобто не є пустою, то впорядкований за зростанням індексу i набір вхідних контактів $X_i^j \in \left[X^{jk}\right]$ позначається ξ_p^j та називається вхідним штекером елемента e_j.

Якщо множина $\left[X^{jk}\right] = 0$, то відповідний штекер не існує.

Число вхідних штекерів ξ_p^j дорівнює числу джерел зовнішнього середовища і елементів системи S, від яких елемент e_j приймає вхідні сигнали.

Набір вхідних штекерів елемента e_j позначається як $\left[\xi_p^j\right]$.

Нехай ξ_p^j – деякий вхідний штекер елемента e_j. Для кожного контакту $X_i^j \in \xi_p^j$ визначимо $Y_l^k = R\left(X_i^j\right)$. Всі вихідні контакти Y_l^k, отримані за допомогою оператора сполучення *R*, згідно з визначенням вхідного штекера належать до одного і того ж елемента e_k.

Упорядкований набір усіх вихідних контактів Y_l^k, пов'язаних елементарними каналами з вхідними контактами штекера ξ_p^j, позначаються η_q^k і називаються вихідними штекерами елемента e_k.

Число вихідних штекерів η_q^k елемента e_k дорівнює сумі числа елементів e_j системи *S* і числа споживачів зовнішнього середовища, яким елемент передає свої вихідні сигнали.

Набір вихідних штекерів елемента e_k позначається як $\left[\eta_q^k\right]$. Всі відомості про штекери системи *S* і зовнішнього середовища заносяться в спеціальну таблицю.

Нехай, наприклад, є система $S = \left\{e_j \mid j \in \overline{1,4}\right\}$ із зовнішнім середовищем з двох джерел $\hat{e}_{01}, \hat{e}_{02}$ і двох споживачів $\tilde{e}_{01}, \tilde{e}_{02}$. Зв'язки між елементами системи і зовнішнього середовища показано на рис. 2.16.

Побудуємо таблицю з інформацією про штекери всіх елементів системи і зовнішнього середовища, показаних на рис. 2.16. Елементи зовнішнього середовища $\hat{e}_{01}, \hat{e}_{02}$, які є джерелом сигналу, мають тільки вихідні штекери. Зокрема, елемент $\hat{e}_{01}$ має один вихідний штекер $\eta_1^{01} = \left\{Y_1^{01}\right\}$ для зв'язку з елементом системи e_1, а елемент $\hat{e}_{02}$ має два вихідних штекера:

- $\eta_1^{02} = \left\{Y_1^{02}\right\}$ для зв'язку з елементом системи e_1;
- $\eta_2^{02} = \left\{Y_2^{02}\right\}$ для зв'язку з елементом системи e_2.

Елементи системи e_1, e_2, e_3, e_4 мають як вхідні, так і вихідні штекери.

Побудуємо таблицю з інформацією про штекери всіх елементів системи і зовнішнього середовища, показаних на рис. 2.16. Елементи зовнішнього середовища $\hat{e}_{01}, \hat{e}_{02}$, які є джерелом сигналу, мають тільки вихідні штекери. Зокрема, елемент $\hat{e}_{01}$ має один вихідний штекер $\eta_1^{01} = \left\{Y_1^{01}\right\}$ для зв'язку з елементом системи e_1, а елемент $\hat{e}_{02}$ має два вихідних штекери:

- $\eta_1^{02} = \left\{Y_1^{02}\right\}$ для зв'язку з елементом системи e_1;
- $\eta_2^{02} = \left\{Y_2^{02}\right\}$ для зв'язку з елементом системи e_2.

Елементи системи e_1, e_2, e_3, e_4 мають як вхідні, так і вихідні штекери.

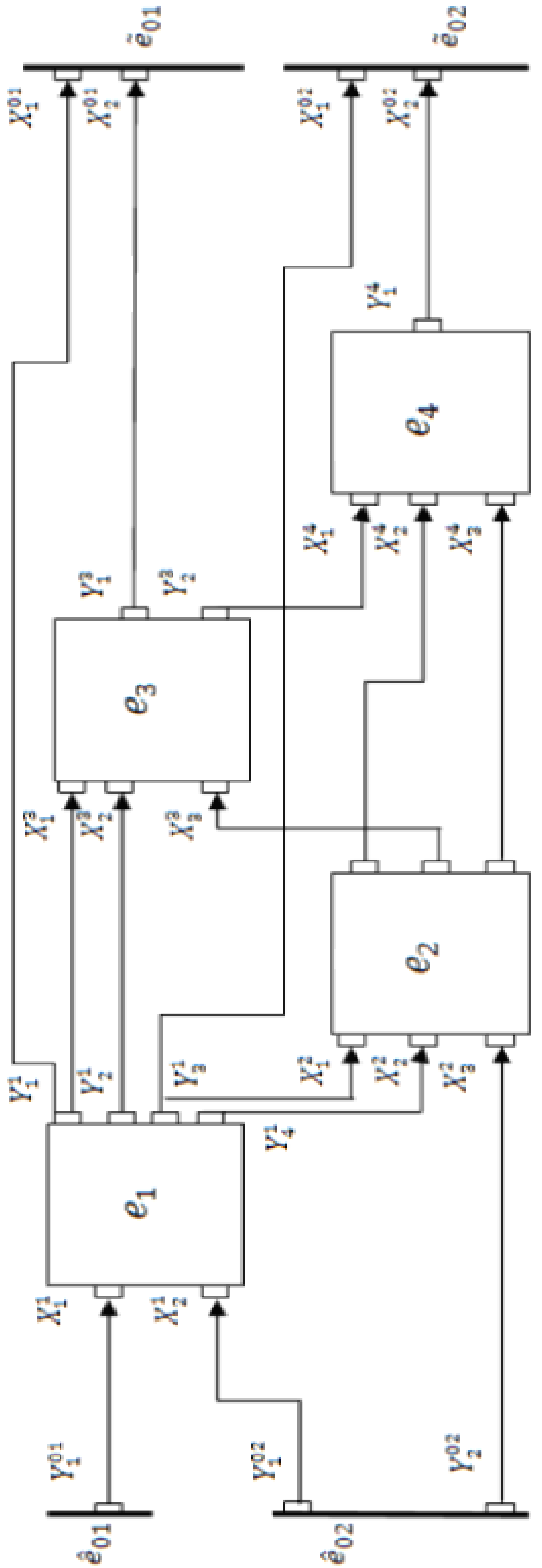


Рисунок 2.16 – Приклад системи S і зовнішнього середовища, елементи яких утворюють штекери

Елемент e_1 має два вхідних штекери:

- $\xi_1^1 = \{X_1^1\}$ для зв'язку із зовнішнім джерелом $\hat{e}_{01}$;
- $\xi_2^1 = \{X_2^1\}$ для зв'язку із зовнішнім джерелом $\hat{e}_{02}$

і чотири вихідних штекери:

- $\eta_1^1 = \{Y_3^1, Y_4^1\}$ для зв'язку з елементом e_2;
- $\eta_2^1 = \{Y_1^1, Y_2^1\}$ для зв'язку з елементом e_3;
- $\eta_3^1 = \{Y_1^1\}$ для зв'язку із зовнішнім споживачем $\tilde{e}_{01}$;
- $\eta_4^1 = \{Y_3^1\}$ для зв'язку із зовнішнім споживачем $\tilde{e}_{02}$.

Елемент e_2 має два вхідних штекери:

- $\xi_1^2 = \{X_3^2\}$ для зв'язку із зовнішнім джерелом $\hat{e}_{02}$;
- $\xi_2^2 = \{X_1^2, X_2^2\}$ для зв'язку з елементом e_1

і два вихідних штекери:

- $\eta_1^2 = \{Y_2^2\}$ для зв'язку з елементом e_3;
- $\eta_2^2 = \{Y_1^2, Y_3^2\}$ для зв'язку з елементом e_4.

Елемент e_3 має два вхідних штекери:

- $\xi_1^3 = \{X_1^3, X_2^3\}$ для зв'язку з елементом e_1;
- $\xi_2^3 = \{X_3^3\}$ для зв'язку з елементом e_2

та два вихідних штекера:

- $\eta_1^3 = \{Y_2^3\}$ для зв'язку з елементом e_4;
- $\eta_2^3 = \{Y_1^3\}$ для зв'язку із зовнішнім споживачем $\tilde{e}_{01}$.

Елемент e_4 має два вхідних штекери:

- $\xi_1^4 = \{X_2^4, X_3^4\}$ для зв'язку з елементом e_2;
- $\xi_2^4 = \{X_1^4\}$ для зв'язку з елементом e_3

та один вихідний штекер $\eta_1^4 = \{Y_1^4\}$ для зв'язку із зовнішнім споживачем $\tilde{e}_{02}$.

Елементи зовнішнього середовища $\tilde{e}_{01}, \tilde{e}_{02}$ є споживачами сигналів і тому мають тільки вхідні штекери.

Елемент $\tilde{e}_{01}$ має два вхідних штекери:

- $\xi_1^{01} = \{X_1^{01}\}$ для зв'язку з елементом e_1;
- $\xi_1^{02} = \{X_2^{02}\}$ для зв'язку з елементом e_4.

Елемент $\tilde{e}_{02}$ має два вихідних штекери:

- $\xi_1^{02} = \{X_1^{02}\}$ для зв'язку з елементом e_1;
- $\xi_2^{02} = \{X_2^{02}\}$ для зв'язку з елементом e_4.

Усі зазначені відомості про склад штекерів заносяться у спеціальну табл. 2.12, рядки якої ідентифікуються номерами штекерів, а стовпці – номерами елементів і зовнішнього середовища.

У клітинах табл. 2.12 на перетині рядків і стовпців записуються індекси контактів, що утворюють штекери. При цьому у верхній частині клітини записуються контакти, що утворюють вхідні штекери, а в нижній частині – контакти, що утворюють вихідні штекери.

Таблиця 2.12 – Відомості про склад штекерів

Номер штекера	Елементи							
	$\widehat{01}$	$\widehat{02}$	1	2	3	4	$\widetilde{01}$	$\widetilde{02}$
1	—	—	(1,1)	(2,3)	(3,1); (3,2)	(4,2); (4,3),	(01,1)	(02,1)
	(01,1)	(02,1)	(1,3); (1,4)	(2,2)	(3,2)	(4,1)	—	—
2	—	—	(1,2)	(2,1); (2,2)	(3,3)	(4,1)	(01,2)	(02,2)
	—	(02,2)	(1,1); (1,2)	(2,1); (2,3)	(3,1)	—	—	—
3	—	—	—	—	—	—	—	—
	—	—	(1,1)	—	—	—	—	—
4	—	—	—	—	—	—	—	—
	—	—	(1,3)	—	—	—	—	—

Інформація про склад штекерів дозволяє перейти до побудови однорівневої схеми сполучення для системи *S* у натуральній формі.

Для реалізації побудови необхідно визначити набори вхідних і вихідних штекерів $\left[\xi_p^j\right], \left[\eta_q^k\right]$, де *j*, *k* – індекси елементів системи і зовнішнього середовища, *p*, *q* – індекси (номери) штекерів, що входять в набір для розглянутого елемента.

Так для прикладу системи *S* і зовнішнього середовища на рис. 2.16 набори штекерів мають вигляд:

$$\left[\xi_p^1\right] = \left\{\xi_1^1, \xi_2^1\right\}, \quad \left[\eta_q^1\right] = \left\{\eta_1^1, \eta_2^1, \eta_3^1, \eta_4^1\right\},$$

$$\left[\xi_p^2\right] = \left\{\xi_1^2, \xi_2^2\right\}, \quad \left[\eta_q^2\right] = \left\{\eta_1^2, \eta_2^2\right\},$$

$$\left[\xi_p^3\right] = \left\{\xi_1^3, \xi_2^3\right\}, \quad \left[\eta_q^3\right] = \left\{\eta_1^3, \eta_2^3\right\},$$

$$\left[\xi_p^4\right] = \left\{\xi_1^4, \xi_2^4\right\}, \quad \left[\eta_q^4\right] = \left\{\eta_2^4\right\},$$

$$\left[\xi_p^{01}\right] = \left\{\xi_1^{01}, \xi_2^{01}\right\}, \quad \left[\eta_q^{01}\right] = \left\{\eta_1^{01}\right\},$$

$$\left[\xi_p^{02}\right] = \left\{\xi_1^{02}, \xi_2^{02}\right\}, \quad \left[\eta_q^{02}\right] = \left\{\eta_1^{02}, \eta_2^{02}\right\}.$$

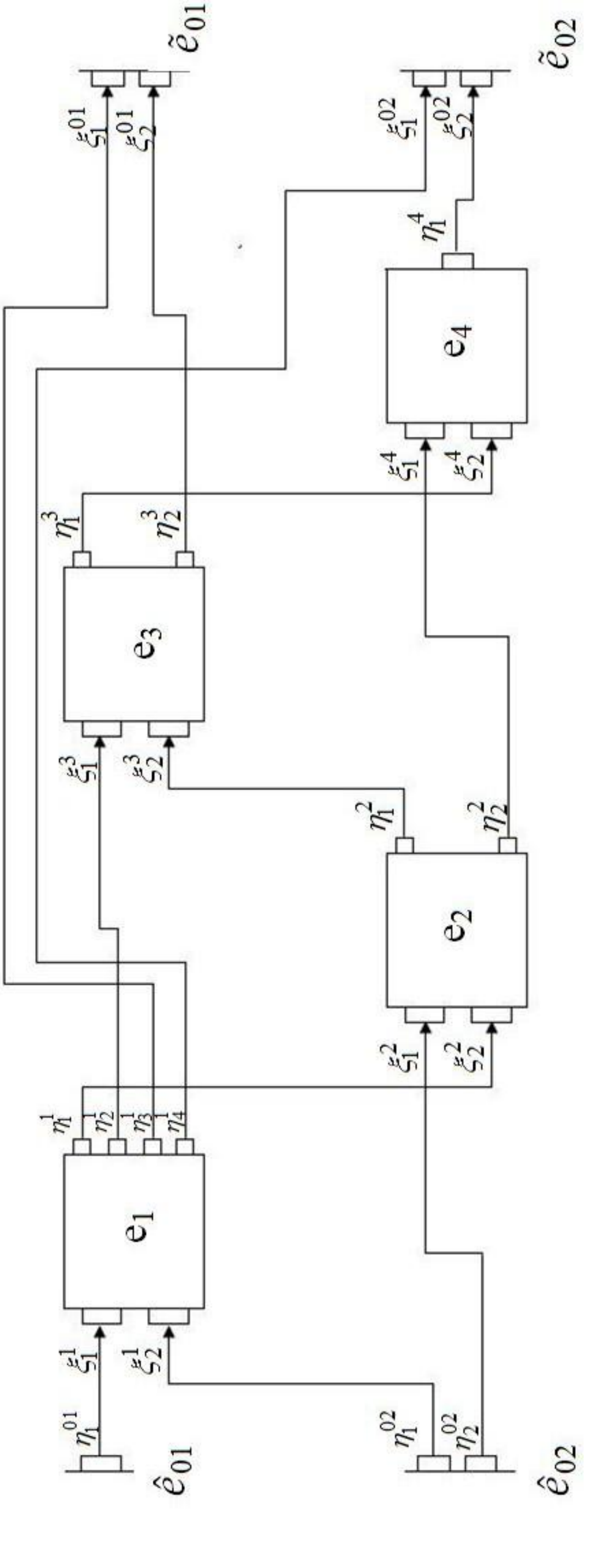


Рисунок 2.17 – Схема сполучення елементів системи і зовнішнього середовища в натуральній формі

Оператором сполучення елементів системи S і зовнішнього середовища в натуральній формі називається оператор $\eta_q^k = \rho\left(\xi_p^j\right)$, який кожному вхідному штекеру ξ_p^j ставить у відповідність єдиний вихідний штекер η_q^k, з'єднаний з ним каналом, що є в загальному випадку сукупністю елементарних каналів, з'єднуючих контакти, які утворюють зазначені штекери.

Оператор сполучення ρ задається за допомогою таблиці, структура і правила заповнення якої аналогічні таблиці для завдання оператора сполучення R. Зокрема, таблиця для оператора сполучення ρ в розглянутому прикладі системи і зовнішнього середовища на рис. 2.16 має вигляд табл. 2.13. Схема сполучення, визначена оператором сполучення, заданого табл. 2.13, показана на рис. 2.17.

Таблиця 2.13 – Приклад завдання оператора сполучення ρ

Штекер	Елементи системи та зовнішнього середовища					
	1	2	3	4	01	02
1	01,1	02,2	1,2	2,2	1,3	1,4
2	02,1	1,1	2,1	3,1	3,2	4,1

Схеми сполучення на рис.2.16 і 2.17 є схемами сполучення тієї ж системи і зовнішнього середовища. Їх порівняння дозволяє зробити висновок, що схема сполучення в натуральній формі на рис. 2.17 трохи простіша від схеми сполучення в канонічній формі на рис. 2.16. Спрощення досягається тим, що число штекерів, а значить, і число зв'язків на рис. 2.17 у елементів системи менше числа контактів на рис. 2.16. Це спрощення є головним обґрунтуванням доцільності використання схем сполучення в натуральній формі для аналізу зв'язків складних систем.

Сукупність наборів вхідних $\left[\xi_p^j\right]$ і вихідних $\left[\eta_q^k\right]$ штекерів разом з таблицею штекерів і оператором сполучення ρ називають моделлю сполучення у вигляді однорівневої схеми сполучення елементів системи S і зовнішнього середовища в натуральній формі.

Процедура побудови моделі сполучення у вигляді багаторівневої схеми сполучення в натуральній формі аналогічна процедурі побудови моделі сполучення у вигляді багаторівневої схеми сполучення в канонічній формі.

Висновки

У цьому розділі розглянуто аналітичне моделювання процесів, що протікають у складних системах. Були виділені наступні види процесів у динамічних системах: неперервно-детерміновані, дискретно-детерміновані, дискретно-стохастичні і неперервно-стохастичні. Для їх опису використовуються типові математичні схеми, до яких належать скінченні автомати, імовірнісні автомати, диференціальні рівняння та системи масового обслуговування. Характер процесу в системі дозволяє визначити вид моделі і вибрати відповідну математичну схему для її опису.

Завдання

1. Розглянемо роботу світлофора на перехресті. Визначте вид моделі і виберіть математичну схему. Обґрунтуйте свій вибір і побудуйте модель досліджуваного об’єкта.
2. Розглянемо роботу банку з одним касиром. Припустимо, що клієнти приходять у банк. Якщо касир зайнятий, вони очікують в одній черзі. Клієнти обслуговуються в порядку прибуття. Після закінчення обслуговування вони йдуть. Визначте вид моделі і виберіть математичну схему. Обґрунтуйте свій вибір і побудуйте модель досліджуваного об’єкта.
3. Розглянемо роботу товарного складу. Всі види товарів розташовуються на своїх полицях. Вміст полиць змінюється при переміщенні товарів. Це може відбуватися під час надходження товару або його відвантаження. Визначте вид моделі і виберіть математичну схему. Обґрунтуйте свій вибір і побудуйте модель досліджуваного об’єкта.
4. Розглянемо просту телефонну систему з двома вихідними лініями зв’язку. Якщо обидві лінії зайняті, дзвінок ставиться в чергу очікування. Дзвінки обслуговуються в порядку надходження. Максимальний час очікування становить 4 хвилини. Визначте вид моделі і виберіть математичну схему. Обґрунтуйте свій вибір і побудуйте модель досліджуваного об’єкта.
5. Розглянемо роботу супермаркету. Покупці можуть брати візок. Кожен візок заблокований для пересування, поки покупець не внесе монету номіналом 25 або 50 копійок. У цьому випадку візок автоматично буде розблоковано. В один момент часу покупець може внести тільки одну монету. Визначте вид моделі і виберіть математичну схему. Обґрунтуйте свій вибір і побудуйте модель досліджуваного об’єкта.

6. Розглянемо роботу CD-ROM, що читає аудіо CD диски. Щоб прослухати трек, необхідно вставити диск і натиснути кнопку. Визначте вид моделі і виберіть математичну схему. Обґрунтуйте свій вибір і побудуйте модель досліджуваного об’єкта.
7. Розглянемо роботу мікрохвильової печі. Для приготування піци потрібно відкрити двері, покласти піцу, закрити двері і натиснути кнопку «старт». Коли піца приготовлена, мікрохвильова піч подає сигнал. Визначте вид моделі і виберіть математичну схему. Обгрунтуйте свій вибір і побудуйте модель досліджуваного об’єкта.
8. Розглянемо роботу ліфта у триповерховому будинку. Визначте вид моделі і виберіть математичну схему. Обґрунтуйте свій вибір і побудуйте модель досліджуваного об’єкта.
9. Розглянемо роботу бензозаправки з двома колонками. Автомобілі обслуговуються в порядку прибуття. Визначте вид моделі і виберіть математичну схему. Обґрунтуйте свій вибір і побудуйте модель досліджуваного об’єкта.

Контрольні запитання

1. Дайте визначення скінченного автомата.
2. Які параметри необхідно визначити для опису скінченного автомата?
3. Які існують види автоматів?
4. Чим відрізняються автомати Мілі та Мура?
5. Коли використовуються синхронні і асинхронні автомати?
6. Як задається імовірнісний автомат?
7. Дайте визначення імовірнісного автомата.
8. Чому необхідно враховувати випадкові фактори при описі досліджуваного об’єкта?
9. Чим відрізняються скінченний та імовірнісний автомати?
10. Як описуються переходи в автоматі з випадковими переходами?
11. Як описуються виходи в автоматі з випадковими виходами?
12. Як використовується D-схема?
13. У яких випадках використовуються детерміновані неперервні моделі?
14. Наведіть приклади моделювання з використанням диференціальних рівнянь.
15. У яких областях для моделювання використовуються D-схеми?
16. Дайте визначення диференціального рівняння.
17. Який вид динамічних систем описується за допомогою теорії масового обслуговування?
18. Які існують дисципліни в черзі?

19. За допомогою яких параметрів описується черга?
20. Які характеристики описують надходження заявок у системі масового обслуговування?
21. Які характеристики описують процес обслуговування?
22. Які характеристики описують чергу в системі масового обслуговування?

3. СТАТИСТИЧНЕ МОДЕЛЮВАННЯ

3.1. Суть методу статистичного моделювання та основні напрями його використання

Суть методу статистичного моделювання, який ще називають методом статистичних випробувань Монте-Карло, полягає в розробленні імітаційної моделі процесу функціонування досліджуваного об'єкта при випадкових вхідних впливах, при випадкових змінах внутрішніх параметрів. Очевидно, що метод статистичного моделювання є складовою частиною тих математичних схем, на основі яких розробляється математична модель для вивчення стохастичних об'єктів. До цих математичних схем належать перш за все розглянуті раніше імовірнісні автомати, системи масового обслуговування та агрегати. Це основний напрямок використання методу, який демонструється наступним прикладом.

Нехай є деяка система S, на вхід якої подаються випадкові вхідні дії x_1 і x_2, як це показано на рис. 3.1.

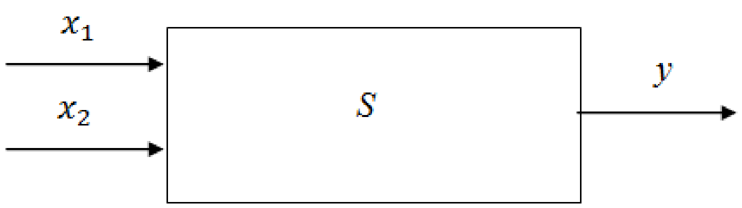


Рисунок 3.1 – Приклад системи з випадковими вхідними впливами

Відомо, що $x_1 = 1 - e^{-\lambda}$, $x_2 = 1 - e^{-\mu}$, де λ, μ – випадкові величини, для яких відома функція їх розподілу (інтегральний закон розподілу).

Відомо також , що $y = \sqrt{x_1^2 + x_2^2}$. Очевидно, що y є випадковою величиною.

Необхідно дати оцінку математичному очікуванню випадкової величини y, тобто $M[y]$. Реалізація методу статистичного моделювання для оцінки значення $M[y]$ зводиться до реалізації імітаційної моделі у вигляді наступного алгоритму:

На основі заданих законів розподілу випадкових величин λ та μ генеруються значення цих випадкових величин λ_i, μ_i $(i \in \overline{1,N})$, де N – кількість значень або обсяг вибірки значень випадкових величин.

Для кожного набору значень λ_i та μ_i обчислюємо значення вихідної величини y:

$$\forall i \in \overline{1,N} \quad y_i = \sqrt{(1-e^{-\lambda_i})^2 + (1-e^{-\mu_i})^2}.$$

На основі множини значень $\{y_i \mid i \in \overline{1,N}\}$ визначається середнє значення випадкових величин y

$$\overline{y} = \frac{1}{N} \sum_{i \in \overline{1,N}} y_i .$$

При $N \to \infty$ отримане значення $\overline{y}$ прагне до оцінки математичного очікування, що є

$$\lim_{N \to \infty} \overline{y} = M[y].$$

Іншим напрямком використання методу є розв'язання детермінованих задач, де випадкові фактори не враховуються. Прикладом цього напряму може бути розв'язання задачі визначення площі під кривою $F(x)$ на інтервалі $0 \le x \le 1$. При цьому $0 \le F(x) \le 1$.

Геометрична інтерпретація розв'язання сформульованої задачі показана на рис. 3.2.

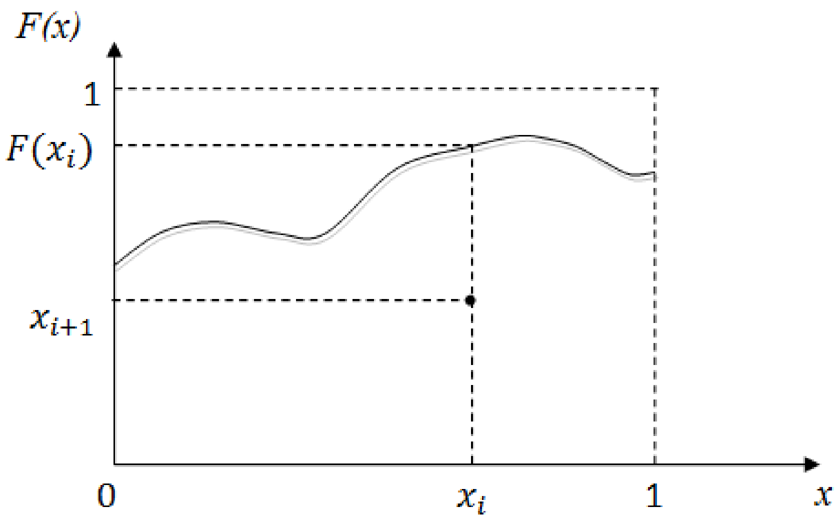


Рисунок 3.2 – Функція $F(x)$ на інтервалі $0 \le x \le 1$

Для розв'язання детермінованих задач методом статистичного моделювання необхідно побудувати стохастичну систему, вихідні характеристики якої при збільшенні числа імітацій N прагнутимуть до розв'язку детермінованої задачі.

Варіантом стохастичної системи для розв'язання детермінованої задачі визначення площі під кривою $F(x)$ методом статистичного моделювання може бути наступний.

Генерується пара незалежних випадкових величин (x_i, x_{i+1}) на інтервалі (0,1). Ця пара може розглядатися як аргумент і як значення функції $F(x)$ відповідно або інакше – як абсциса і ордината функції $F(x)$ на координатній площині рис. 3.2.

Якщо точка (x_i, x_{i+1}), як це показано на рис. 3.2, належить площі під кривою $F(x)$, то виробляється сигнал $h_i = 1$. В іншому випадку, $h_i = 0$.

Це формалізується виразом

$$h_i = \begin{cases} 1, \text{ якщо } x_{i+1} \le F(x_i), \\ 0, \text{ якщо } x_{i+1} > F(x_i). \end{cases}$$

Обчислюється значення

$$\tilde{S} = \frac{1}{N} \sum_{i \in \overline{1,N}} h_i .$$

Тобто значення $\tilde{S}$ прагне до значення площі S, що визначається, при збільшенні числа N, або $\lim\limits_{N \to \infty} \tilde{S} = S$.

3.2. Псевдовипадкові числа і процедури їх генерації

У методі статистичного моделювання, як правило, використовується генерація випадкових чисел. На практиці генерація реалізується трьома способами: апаратним, або фізичним, табличним або файловим, алгоритмічним, або програмним [6].

Апаратний спосіб полягає в отриманні випадкових чисел за допомогою спеціальної електронної приставки, яку називають генератором, або датчиком, випадкових чисел. Фізичною основою отримання випадкових чисел за допомогою датчика є шуми в електронних і напівпровідникових приладах.

Процес отримання послідовності випадкових чисел апаратним способом реалізується відповідно до схеми на рис. 3.3.

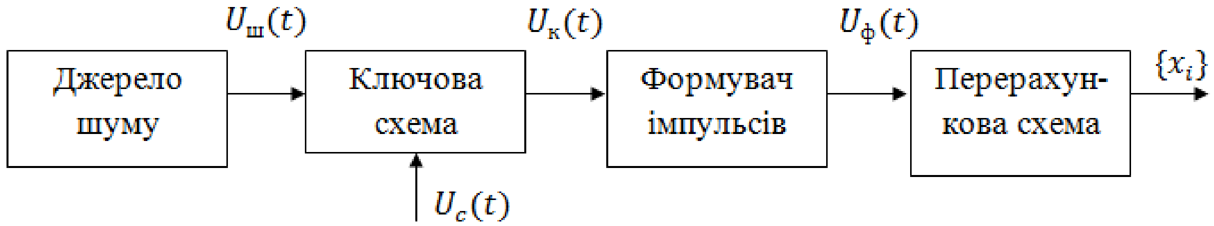


Рисунок 3.3 – Структурна схема формування послідовності випадкових чисел апаратним способом

На рис. 3.3 використано позначення:

$U_m(t)$ – напруга шуму на виході електронного або напівпровідникового приладу;

$U_c(t)$ – амплітуда імпульсу на інтервалі (0, T);

$U_k(t)$ – відрізок шуму на інтервалі (0, T), значення амплітуди якого зіставляються зі значенням амплітуди імпульсу;

$U_ф(t)$ – серія імпульсів у моменти часу t_i $(i \in \overline{1,N})$ на інтервалі [0, T], що виникає при збігу значень амплітуди шуму і значень амплітуди імпульсу;

$\{x_i\}$ – послідовність випадкових чисел з інтервалу (0,1).

Перетворення напруги шуму $U_m(t)$ у послідовність імпульсів, що виникають у моменти t_i із інтервалу [0,T], показано на рис. 3.4.

Моменти виникнення імпульсів t_i фіксуються і на їх основі виводяться значення

$$x_i = \frac{t_i' - t_i}{T}, \text{ де } t_i' \geq t_i, \quad t_i, t_i' \in (0, T),$$

множина яких $\{x_i\}$ і утворює множину випадкових чисел з інтервалу (0,1).

Основними недоліками апаратного способу є неможливість повторення отриманої послідовності випадкових величин, що іноді необхідно для проведення повторного експерименту, і необхідність забезпечення стабільності роботи датчика випадкових величин. Якщо стабільність не забезпечується, то можуть виникнути ситуації, коли $U_c(t) > \max\limits_{t \in [0,T]} U_m(t)$ або коли $U_c(t) < \min\limits_{t \in [0,T]} U_m(t)$. Легко зрозуміти, що в зазначених ситуаціях є неможливим перетин амплітуди $U_m(t)$ та амплітуди $U_c(t)$ на інтервалі (0,T), а значить неможливим є отримання імпульсів на інтервалі (0, T).

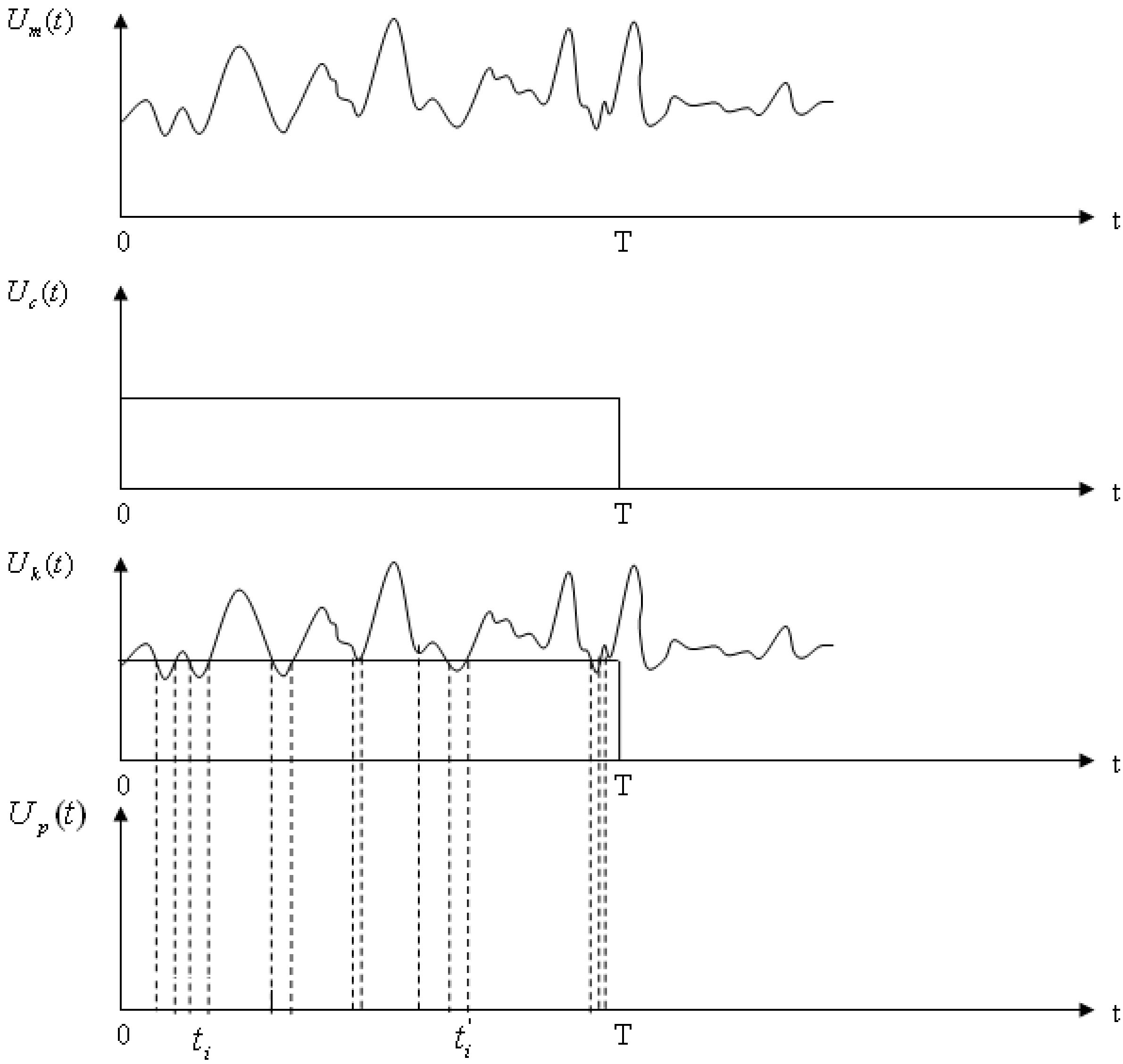


Рисунок 3.4 – Приклад накладення графіка $U_m(t)$ на $U_c(t)$ для отримання послідовності імпульсів

Табличний, або файловий спосіб полягає в тому, що попередньо сформовані випадкові числа оформляються у вигляді таблиці, яку поміщають у книгу або в пам'ять комп'ютера, перед цим перетворивши цю послідовність випадкових чисел у відповідний файл.

Недоліки табличного, або файлового способу досить очевидні і полягають головним чином в обмеженості послідовності випадкових чисел і в нераціональному використанні пам'яті комп'ютера.

Алгоритмічний, або програмний спосіб реалізується комп'ютером на основі спеціальних алгоритмів і програм. Цей спосіб дозволяє отримати послідовність випадкових величин, що підпорядковується будь-яким заданим законам розподілу. Така можливість забезпечується шляхом функціонального перетворення базової послідовності випадкових величин, у якості якої розглядається послідовність випадкових величин, рівномірно розподілених на інтервалі (0,1).

Строго кажучи, алгоритмічний спосіб дозволяє отримати послідовність псевдовипадкових величин, принаймні, з двох причин. По-перше, комп’ютер оперує зі скінченною множиною чисел і, по-друге, використання алгоритмів вносить порядок, закономірність, тобто детермінованість.

Широке розповсюдження отримали конгруентні процедури генерації послідовності псевдовипадкових величин.

Два числа a і b конгруентні (порівняні) за модулем m, де m – ціле позитивне число, тоді і тільки тоді, коли існує таке ціле позитивне число k, що $a-b=k\cdot m$, і якщо числа a і b дають однакові залишки при їх діленні на m.

Наприклад, числа a=375 і b=125 є конгруентними за модулем m = 10.

Дійсно, $a-b=375-125=250=25\cdot 10$, тобто k=25, m=10.

При цьому 5 є однаковим залишком від ділення a = 375 і b = 125 на m = 10, що записується у вигляді $5\equiv 375(\mathrm{mod}\,10)$, $5\equiv 125(\mathrm{mod}\,10)$.

Конгруентні процедури мають детермінований характер і задаються рекурентним співвідношенням

$$x_{i+1}\equiv \lambda x_i+\mu(\mathrm{mod}\,M) \tag{3.1}$$

де x_i,λ,μ,M – невід’ємні цілі числа.

Розкриємо співвідношення (3.1), вважаючи спочатку i = 0.

$$x_1\equiv \lambda x_0+\mu(\mathrm{mod}\,M).$$

Далі нехай i = 1.

$$x_2\equiv \lambda x_1+\mu(\mathrm{mod}\,M)\equiv \lambda(\lambda x_0+\mu)+\mu(\mathrm{mod}\,M)\equiv \lambda^2 x_0+\mu(\lambda+1)(\mathrm{mod}\,M)\equiv$$

$$\equiv \lambda^2 x_0+\mu\frac{\lambda^2-1}{\lambda-1}(\mathrm{mod}\,M)$$

Припустимо i = 2.

$$x_3=\lambda x_2+\mu(\mathrm{mod}\,M)\equiv \lambda(\lambda^2 x_0+\mu\frac{\lambda^2-1}{\lambda-1})+\mu(\mathrm{mod}\,M)\equiv$$

$$\equiv \lambda^3 x_0+\mu(\lambda^2+\lambda+1)(\mathrm{mod}\,M)\equiv \lambda^3 x_0+\mu\frac{\lambda^3-1}{\lambda-1}(\mathrm{mod}\,M).$$

Закономірність має наступний вигляд:

$$x_i\equiv \lambda^i x_0+\mu\frac{\lambda^i-1}{\lambda-1}(\mathrm{mod}\,M) \tag{3.2}$$

Таким чином, якщо відомі значення x_0, λ, μ, M, то вираз (3.2) визначає послідовність цілих невід'ємних чисел $\{x_i\}$, складену із залишків поділу членів послідовності $\left\{\lambda^i x_0 + \mu \dfrac{\lambda^i - 1}{\lambda - 1}\right\}$ на значення *M*.

Оскільки $x_i < M$, то на основі послідовності $\{x_i\}$ може бути побудована послідовність $\{x_i\} = \{x_i / M\}$, яка є послідовністю раціональних чисел з одиничного інтервалу (0,1).

Конгруентну процедуру, що задається виразом (3.2), називають змішаним методом генерації псевдовипадкових чисел, квазірівномірно розподілених в інтервалі (0,1).

Якщо у виразі (3.2) припустити $\mu = 0$, то отримуємо вираз, що є основою мультиплікативного методу генерації псевдовипадкових чисел квазірівномірно розподілених в інтервалі (0,1).

Скористаємося мультиплікативним методом для отримання псевдовипадкових квазірівномірно розподілених чисел в інтервалі (0,1), вважаючи $\lambda = 3, x_0 = 4, M = 11$.

Метод реалізується послідовністю наступних кроків:

1. Формується послідовність чисел

$$\{\lambda^i x_0\} = \{\lambda x_0, \lambda^2 x_0, \lambda^3 x_0, \lambda^4 x_0, ...\} = \{12, 36, 108, 324, 972, 2916, 8748, ...\}$$

2. Формується послідовність чисел $\{x_i\}$, як послідовність із залишків від ділення чисел послідовності $\{\lambda^i x_0\}$ на *M* при відповідних значеннях *i*. Ця послідовність має вигляд

$$\{x_i\} = \{1, 3, 9, 5, 4, 1, 3...\} \tag{3.3}$$

3. Формується послідовність раціональних чисел з множини (0,1). Ця послідовність для розглянутого прикладу має вигляд:

$$\{x_i\} = \{\frac{1}{11}, \frac{3}{11}, \frac{9}{11}, \frac{5}{11}, \frac{4}{11}, \frac{1}{11}, \frac{3}{11}, ...\} = \{0.09, 0.27, 0.81, 0.45, 0.36, 0.09, 0.27, ...\}.$$

Якість одержуваних псевдовипадкових квазірівномірно розподілених чисел перевіряється за допомогою тестів на рівномірність, стохастичність і незалежність.

Перевірка на рівномірність може бути виконана за допомогою гістограми на рис. 3.5.

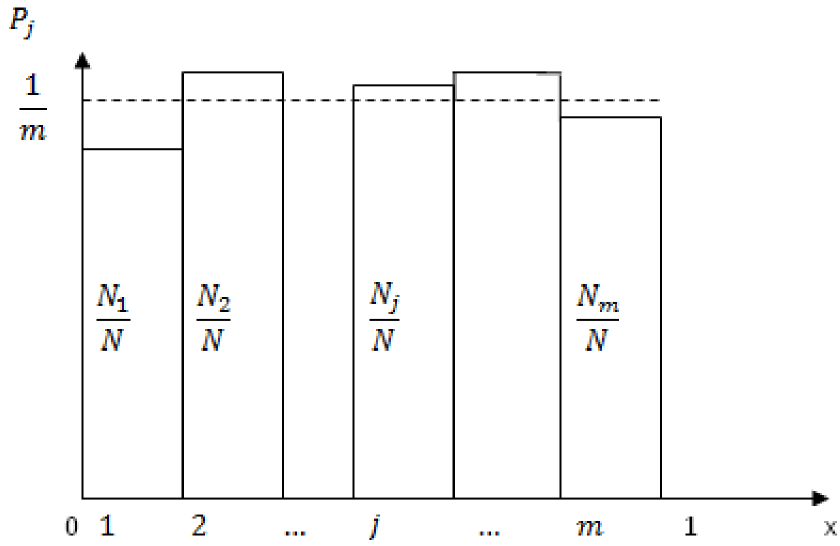


Рисунок 3.5 – Гістограма для перевірки послідовності на рівномірність

Для перевірки інтервал (0,1) розбивається на m рівних частин ($m = 20 \div 50$). Кожна частина утворює підінтервал, який ідентифікується індексом $j \in \overline{1,m}$.

Висувається гіпотеза про рівномірність розподілу чисел послідовності $\{x_i\}$ в інтервалі (0,1). Якщо гіпотеза вірна, то кожне з чисел x_i з імовірністю $p_j = \dfrac{1}{m}$ потрапляє на j-й підінтервал ($j \in \overline{1,m}$).

Для перевірки правильності висунутої гіпотези числа послідовності $\{x_i\}$ розподіляють по підінтервалах $j \in \overline{1,m}$. Нехай N_j – кількість чисел з послідовності $\{x_i\}$, які потрапили на підінтервал $j \in \overline{1,m}$, а $N = \sum\limits_{j \in \overline{1,m}} N_j$ – загальна кількість чисел послідовності $\{x_i\}$. Відношення N_j / N визначає відносну частоту потрапляння чисел з послідовності $\{x_i\}$ на підінтервал $j \in \overline{1,m}$. Це відношення є ординатою гістограми на j-му підінтервалі, що і показано на рис. 3.5.

Якщо гіпотеза про рівномірність розподілу вірна, то ординати N_j / N ($j \in \overline{1,m}$) експериментальної гістограми будуть близькі до значення l / m. Кількісну оцінку ступеня близькості можна отримати за допомогою одного з критеріїв узгодженості – Стьюдента, Пірсона та ін.

Перевірка на стохастичність часто реалізується методом комбінацій, що зводиться до визначення закону розподілу появи кількості одиниць (нулів) у n-розрядному двійковому поданні числа x_i послідовності $\{x_i\}$. На практиці розглядаються не всі n розрядів числа x_i, а обмежуються старшими l розрядами.

Відповідно до теорії законом розподілу появи j одиниць у l старших розрядах двійкового числа за умови незалежності окремих розрядів є біноміальний закон розподілу, тобто

$$p(j,l) = C_l^j p^j(1)[1-p(1)]^{l-j},$$

де $p(j,l)$ – ймовірність появлення j одиниць у l розрядах; $C_l^j = \dfrac{l!}{j!(l-j)!}$ – число можливих поєднань; $p(1)$ – ймовірність появи одиниці в одному розряді.

Оскільки ймовірність появи одиниці в одному розряді дорівнює ймовірності появи нуля, тобто

$$p(1) = p(0) = [1-p(1)] = 0{,}5,$$

то біноміальний закон розподілу для розглянутого випадку приймає вигляд:

$$p(j,l) = C_l^j p^l(1).$$

Результатом експерименту є отримання послідовності з N чисел $\{x_i\}$. Нехай у цій послідовності n_j чисел з j одиницями в l старших розрядах чисел x_i. Тоді відношення $\dfrac{n_j}{N}$ може розглядатися як частота появи чисел x_i з j одиницями в l старших розрядах.

Висувається гіпотеза про близькість значень $p(j,l)$ і $\dfrac{n_j}{N}$ при різних значеннях $l \le n$. Справедливість гіпотези перевіряється з використанням принципів узгодженості. Якщо гіпотеза вірна, то стохастичність послідовності $\{x_i\}$ вважається справедливою.

Перевірка на незалежність чисел послідовності $\{x_i\}$ реалізується визначенням значень кореляційних моментів $Rx_k x_l$, де $x_k, x_l \in \{x_i\}$, $x_k \ne x_l$. Для незалежних випадкових величин x_k, x_l $Rx_k x_l = 0$ [19].

Важливою характеристикою програмних генераторів є довжина відрізка аперіодичності L. У межах відрізка аперіодичності числа послідовності $\{x_i\}$ не повторюються. Довжина відрізку визначається кількістю чисел, які не повторюються. Наприклад, якщо послідовність $\{x_i\}$ вигляду (3.3) продовжити, то вона набуває такого вигляду:

$$\{x_i\} = \{1,3,9,5.4,1,3,9,5,4,1,3,9,5,4, \ldots\}.$$

У цій послідовності виділяється відрізок з п'яти неповторюваних чисел 1,3,9,5,4. Отже, довжина відрізка аперіодичності в даній послідовності $\{x_i\}$ дорівнює $L=5$.

Довжина L визначається значеннями x_0, M, тобто: $L=L(x_0, M)$ для мультиплікативного методу. Для змішаного методу $L=L(x_0, M, \mu)$.

Якісною особливістю довжини відрізка аперіодичності є те, що використання вибірки з послідовності $\{x_i\}$, довжина якої більша значення L, не призводить до нових статистичних результатів, а вимагає лише додаткових витрат на проведення експериментів.

3.3. Моделювання випадкових впливів

Основним призначенням методу статистичного моделювання є забезпечення досліджень систем різної природи з урахуванням випадкових впливів. До випадкових впливів відносять випадкові події, випадкові дискретні і неперервні величини, випадкові вектори, функції і процеси. Врахувати всі ці види випадкових впливів можна шляхом їх моделювання, для реалізації якого необхідна послідовність псевдовипадкових квазірівномірно розподілених чисел $\{x_i\}$ в інтервалі (0,1), що є базовою послідовністю випадкових чисел.

Моделювання випадкових впливів почнемо з моделювання окремих випадкових подій. Нехай необхідно промоделювати реалізацію окремої випадкової події А, заданої ймовірністю $P(A)$.

Процедура моделювання полягає у виборі з базової випадкової послідовності $\{x_i\}$ деякого значення x_i'.

Якщо виявиться, що $0 < x_i' \le P(A)$, то подія A відбудеться. В іншому випадку, коли $x_i' > P(A)$, то відбудеться подія $\overline{A}$, тобто подія A не відбудеться.

Моделювання випадкової події A_j, що входить до повної групи подій $A_1\ A_2\ A_j\ \ldots\ A_m$, заданих ймовірностями $P(A_1)\ P(A_2)\ \ldots\ P(A_j)\ \ldots\ P(A_m)$, такими, що

$$\sum_{j\in\overline{1,m}} P(A_j) = 1,$$

зводиться до реалізації наступних дій.

З базової послідовності $\{x_i\}$ вибирається деяке значення x_i'.

Якщо $0 < x_i' \le P(A_1)$, то відбудеться подія A_1.

Якщо $P(A_1) < x_i' \le P(A_1) + P(A_2)$, то відбудеться подія A_2.

……………………………………………………………………..

Якщо $\sum_{k\in\overline{1,j-1}} P(A_k) < x_i' \le \sum_{k\in\overline{1,j}} P(A_k)$, то відбудеться подія A_j.

……………………………………………………………………..

Якщо $\sum_{k\in\overline{1,m-1}} P(A_m) < x_i' \le 1$, то відбудеться подія A_m.

Для моделювання складної події, залежної від двох і більше простих подій, заданих ймовірностями виникнення, необхідно спочатку визначити можливі варіанти складної події та ймовірності їх виникнення.

Наприклад, нехай складна подія залежить від двох простих незалежних одна від одної подій *A* і *B* з ймовірностями виникнення *P*(*A*), *P*(*B*). Можливими варіантами складної події в даному випадку можуть бути:

1. спільне виникнення подій *A* і *B* – *AB*;
2. не відбувається подія *A* і відбувається подія *B* – $\overline{A}B$;
3. відбувається подія *A* і не відбувається подія *B* – $A\overline{B}$;
4. не відбувається подія *A* і не відбувається подія *B* – $\overline{A}\overline{B}$.

Визначаються ймовірності зазначених варіантів складної події:

$P_1 = P(A)\,P(B)$;

$P_2 = P(\overline{A})\,P(B) = (1 - P(A))\,P(B)$;

$P_3 = P(A)\,P(\overline{B}) = P(A)\,(1 - P(B))$;

$P_4 = P(\overline{A})\,P(\overline{B}) = (1 - P(A))\,(1 - P(B))$.

Тепер може бути реалізована процедура моделювання варіантів складної події. З послідовності $\{x_i\}$ обирається значення x_i':

Якщо $0 < x_i' \le p_1$, то відбувається варіант *AB*.

Якщо $p_1 < x_i' \le p_1 + p_2$, то відбувається варіант $\overline{A}B$.

Якщо $p_1 + p_2 < x_i' \le p_1 + p_2 + p_3$, то відбувається варіант $A\overline{B}$.

Якщо $p_1 + p_2 + p_3 < x_i' \le p_1 + p_2 + p_3 + p_4$, то відбувається варіант $\overline{A}\overline{B}$.

Зауважимо, що $p_1 + p_2 + p_3 + p_4 = 1$. Це гарантує закінчення процедури і однозначність виникнення того чи іншого варіанта складної події.

Процедура моделювання складної події змінюється у разі, коли події *A* і *B* є залежними. У цьому випадку необхідно знати значення умовних імовірностей. Наприклад, *P*(*A/B*), тобто ймовірність появи події *A* за умови, що подія *B* сталася.

Процедура моделювання в цьому випадку може бути наступною. Нехай відомі значення ймовірності появи події *A* – *P(A)* і умовні ймовірності *P(B/A)*, $P(B/\overline{A})$. З базової послідовності $\{x_i\}$ обирається два значення x_l' и x_k'.

Якщо виявиться, що $0 < x'_l \le P(A)$, то сталася подія *A*. Якщо до того ж виявляється, що $0 < x'_k \le P(B/A)$, то відбувається подія *B*. У цілому виникає варіант складної події *AB*.

Якщо виявиться, що $0 < x'_l \le P(A)$ та $P(B/A) < x'_k \le 1$, то відбувається варіант $A\overline{B}$.

Якщо виявиться, що $P(A) < x'_l \le 1$ та $0 < x'_k \le P(B/\overline{A})$, то відбувається варіант $\overline{A}B$.

І, нарешті, якщо виявиться, що $P(A) < x'_l \le 1$ та $P(B/\overline{A}) < x'_k \le 1$, то відбувається варіант $\overline{A}\overline{B}$.

Дискретно-стохастичне моделювання (підрозділ 2.5) реалізується на основі поняття імовірнісного автомата, у визначенні якого використовується матриця з випадковими переходами. Ця матриця аналогічна матриці переходів однорідного марковського ланцюга:

$$P = \begin{Vmatrix} P_{11} & P_{12} & \dots & P_{1k} \\ P_{21} & P_{22} & \dots & P_{2k} \\ \dots & \dots & \dots & \dots \\ P_{k1} & P_{k2} & \dots & P_{kk} \end{Vmatrix},$$

де P_{ij} – ймовірність переходу зі стану z_i у стан z_j.

При цьому $\forall i \in \overline{1,k} \quad \sum_{j \in \overline{1,k}} P_{ij} = 1$.

Для моделювання однорідного марковського ланцюга з матрицею переходів *P* можна скористатися процедурою жереба, яка застосовується для реалізації функціонування автомата з випадковими переходами або, в більш загальному випадку, для реалізації функціонування імовірнісного автомата [6].

На першому кроці процедури моделювання визначається початковий стан z_0, що задається ймовірностями $p_1(0)$, $p_2(0)$, ... , $p_i(0)$, … , $p_k(0)$, де $p_i(0)\ (i \in \overline{1,k})$ – ймовірність того, що початковим станом буде стан z_i.

Передбачається, що марковський процес є ергодичним. Це означає, що граничний розподіл ймовірностей, який являє собою значення ймовірностей $p_i(n)\ (i \in \overline{1,k})$ находження стану $z_i\ (i \in \overline{1,k})$ через *n* переходів, не залежить від початкового стану, тобто від значення $p_i(0)\ (i \in \overline{1,k})$. Тому значення $p_i(0)\ (i \in \overline{1,k})$ можна вважати такими, що

$$p_1(0) = p_2(0) = \dots = p_k(0) = \frac{1}{k}. \qquad (3.4)$$

Для визначення початкового стану z_0 скористаємося (3.4) і з базової послідовності $\{x_i\}$ виберемо деяке значення x_i'.

Якщо $0 < x_i' \le \frac{1}{k}$, то $z_0 = z_1$, тобто в якості початкового стану вибирається перший стан. Якщо $\frac{1}{k} < x_i' \le \frac{2}{k}$, то $z_0 = z_2$. І так далі. Нарешті, якщо $\frac{k-1}{k} < x_i' \le \frac{1}{k}$, то $z_0 = z_k$.

Нехай для визначеності виявилося, що

$$\frac{l-1}{k} < x_i' \le \frac{l}{k},$$

тобто $z_0 = z_l$.

Вибір стану z_l дозволяє зафіксувати в матриці переходів P індекс рядка $i=l$.

Наступний крок процедури моделювання зв’язується з визначенням стану, в який перейде марківський процес зі стану z_l. Для цього скористаємося розподілом ймовірностей в l-му рядку матриці переходів P:

$$p_{l1}\ p_{l2}\ p_{l3}\ \ldots\ p_{lj}\ \ldots\ p_{lk}.$$

З базової послідовності $\{x_i\}$ вибирається деяке значення x_i''.

Якщо $0 < x_i'' \le p_{l1}$, то процес із стану z_l перейде у стан z_1.

Якщо $p_{l1} < x_i'' \le p_{l1} + p_{l2}$, то процес із стану z_l перейде у стан z_2.

І так далі. У загальному випадку, якщо

$$\sum_{j\in 1,m-1} p_{lj} < x_i'' \le \sum_{j\in 1,m} p_{lj},$$

то процес перейде зі стану z_l у стан z_m.

Визначеність вибору стану z_j, в який перейде процес зі стану z_l, гарантується тим, що сума ймовірностей у кожному рядку матриці переходів дорівнює 1.

На наступному кроці процедури моделювання обраний стан z_m визначає новий індекс $(i=m)$ рядка матриці переходів P, в якій відбивається відповідний розподіл ймовірностей переходів. Цей розподіл використовується так само, як і на попередньому кроці, для вибору наступного стану процесу. І так далі, процедура повторюється необхідну кількість разів, відповідно до числа переходів n.

Моделювання випадкових впливів у вигляді випадкових дискретних і неперервних величин реалізується згідно з методом зворотної функції шляхом перетворення значень випадкової величини ξ з базової послідовності $\{x_i\}$ у значення $\{y_i\}$ випадкової величини η, інтегральний закон розподілу якої $F_\eta(y)$ задано. Формально це перетворення подається як

$$\eta = F_\eta^-(y), \tag{3.5}$$

де F_η^- – функція, зворотна F_η.

Графічно сутність перетворення (3.5) показано на рис. 3.6.

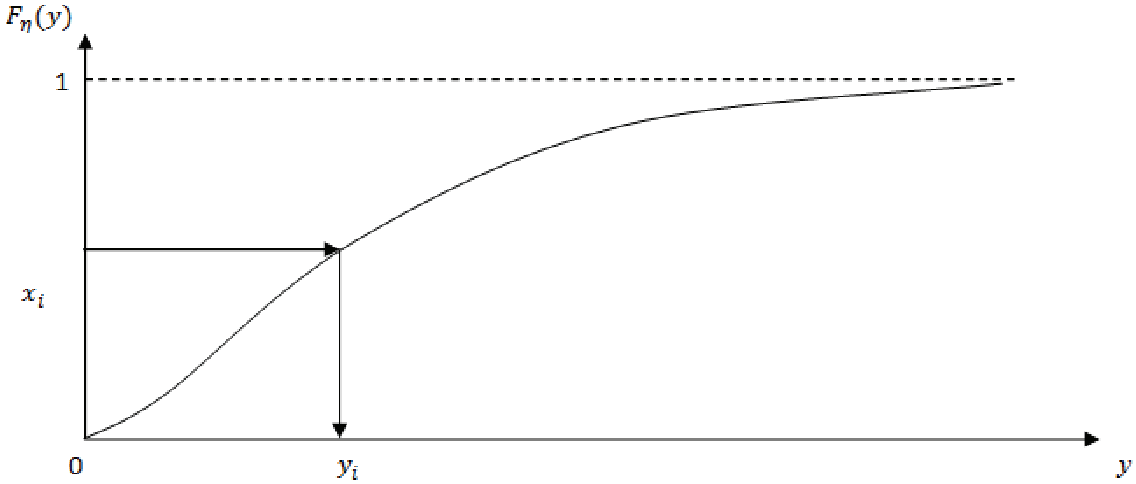


Рисунок 3.6 – Графічна інтерпретація методу зворотної функції

Якщо ξ є рівномірно розподіленою випадковою величиною на інтервалі (0,1), то значення випадкової величини ξ належать базовій послідовності $\{x_i\}$. З послідовності $\{x_i\}$ вибирається деяке значення x_i, яке розглядається як значення функції $F_\eta(y)$ на інтервалі (0,1).

Розв'язується зворотна задача, яка полягає у визначенні аргументу за заданим значенням функції. На рис. 3.6 показується, як за значенням функції $F_\eta(y)$, рівному значенню x_i, визначається значення аргументу y_i.

Нехай інтегральний закон розподілу дискретної випадкової величини η, яка приймає значення $0 < y_1 < y_2 < \ldots < y_m < \ldots$ з ймовірністю $p_1, p_2, \ldots, p_m, \ldots$ визначається як

$$F_\eta(y_m) = P(\eta \le y_m) = \sum_{i \in \overline{1,m}} p_i \quad (m = 1,2,\ldots).$$

При цьому $F_\eta(y_1) = 0$.

Графік закону розподілу $F_\eta(y_m)$ показано на рис. 3.7.

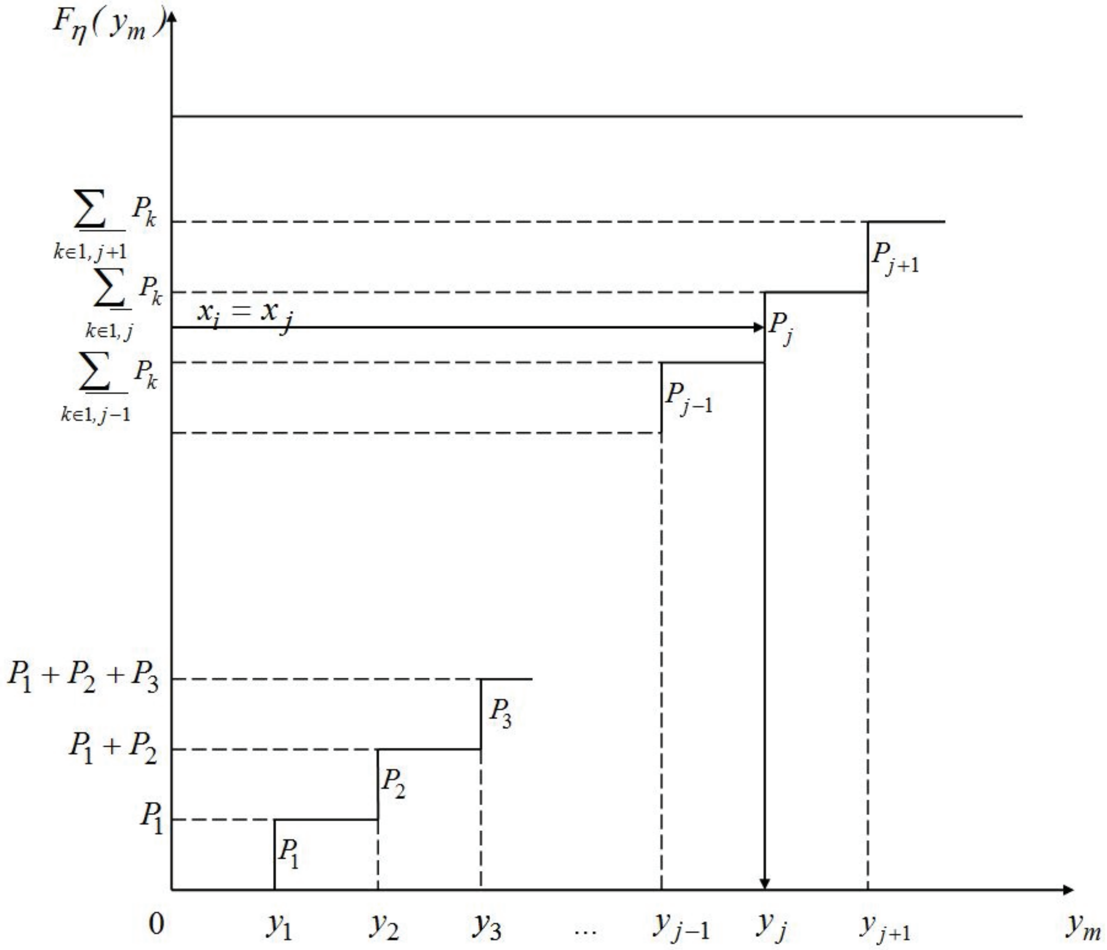


Рисунок 3.7 – Приклад інтегрального закону розподілу дискретної випадкової величини

Значення дискретної випадкової величини $\eta = y_m$ як значення аргументу закону розподілу $F_\eta(y_m)$ залежить від того, в який інтервал значень сум імовірностей $p_i\ (\sum p_i)$ потрапляє значення x_i з базової послідовності $\{x_i\}$, аналізованої як значення функції $F_\eta(y_m)$.

Зокрема, на рис. 3.7 зображено ситуацію, коли значення $x_i = x_j$ задовольняє співвідношенню

$$\sum_{i\in 1,j-1} p_i < x_j \le \sum_{i\in 1,j} p_i .$$

Тому, згідно з методом зворотної функції, відповідним значенням аргументу функції $F_\eta(y_m)$ буде $y_m = y_j$.

Як приклад реалізації методу зворотної функції для отримання послідовності $\{y_i\}$ значень дискретної випадкової величини η розглянемо випадок, коли випадкова величина підпорядковується біноміальному закону розподілу, що задає ймовірність y успішних результатів у N експериментах:

$$P_N(y) = C_N^y p^y (1-p)^{N-y} = \frac{N!}{y!(N-y)!} p^y (1-p)^{N-y},$$

де p – ймовірність успішного результату в одному експерименті.

Нехай для визначеності p=0,5, N=6.

Визначимо спочатку, які значення приймає розглянута дискретна випадкова величина і розподіл імовірностей їх прийняття.

Можливими значеннями є:

$$y_1=0,\ y_2=1,\ y_3=2,\ y_4=3,\ y_5=4,\ y_6=5,\ y_7=6,$$

тобто в шести експериментах (N = 6) можливо нуль, один і так далі успішних результатів.

Для визначення розподілу ймовірностей прийняття вказаних значень дискретної випадкової величини скористаємося виразом біноміального закону розподілу:

$$P_N(y_1) = \frac{N!}{y_1!(N-y_1)!} p^{y_1} (1-p)^{N-y_1} = P_6(0) = \frac{6!}{0!(6-0)!} 0.5^0 (1-0.5)^{6-0} = 0.01562;$$

$P_N(y_2) = P_6(1) = 0.09375$, $P_N(y_3) = P_6(2) = 0.23438$, $P_N(y_4) = P_6(3) = 0.31250$;

$P_N(y_5) = P_6(4) = 0.23438$, $P_N(y_6) = P_6(5) = 0.09375$, $P_N(y_7) = P_6(6) = 0.01562$.

Отримані значення дискретної випадкової величини $\{y_i\}$ і розподілу ймовірностей їх появи дозволяють побудувати графік інтегрального біноміального закону розподілу розглянутого випадку, який показано на рис. 3.8.

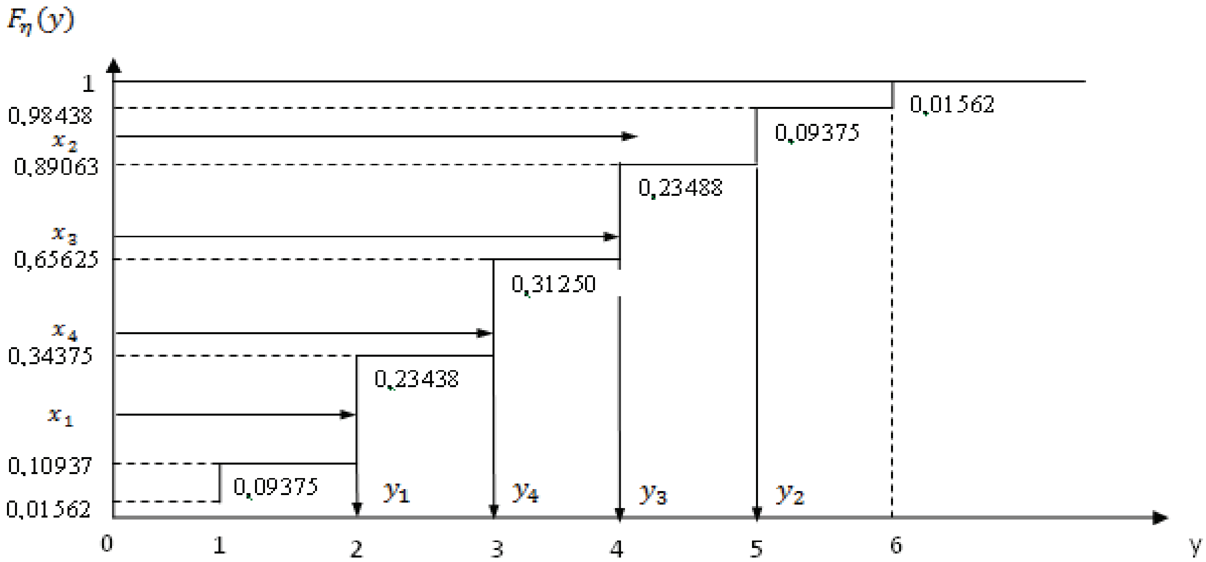


Рисунок 3.8 – Графік інтегрального біноміального закону розподілу і його використання для послідовності $\{y_i\}$

На графіку зображено такі суттєві значення $F_\eta(y_m)$:

$F_\eta(y_1) = P_N(y_1) = p_1 = 0{,}01562$,

$F_\eta(y_2) = P_N(y_1) + P_N(y_2) = p_1 + p_2 = 0{,}10937$,

$F_\eta(y_3) = P_N(y_1) + P_N(y_2) + P_N(y_3) = p_1 + p_2 + p_3 = 0{,}34375$,

$F_\eta(y_4) = P_N(y_1) + P_N(y_2) + P_N(y_3) + P_N(y_4) = p_1 + p_2 + p_3 + p_4 = \quad = 0{,}65625$,

$F_\eta(y_5) = P_N(y_4) + P_N(y_2) + P_N(y_3) + P_N(y_4) + P_N(y_5) = p_1 + p_2 + p_3 + + p_4 + p_5 = 0{,}89063$,

$\mathrm{F}_\eta(y_6) = P_N(y_4) + P_N(y_2) + P_N(y_3) + P_N(y_4) + P_N(y_5) + P_N(y_6) = p_1 + + p_2 + p_3 + p_4 + + p_5 + p_6 = 0{,}98438$.

Указані значення $F_\eta(y_1), F_\eta(y_2), \ldots, F_\eta(y_6)$, визначені як відповідні суми ймовірностей, використовуються для формування послідовності $\{y_i\}$ на основі базової послідовності $\{x_i\}$.

Нехай $\{x_i\}=\{0{,}19; 0{,}94; 0{,}69; 0{,}44; \ldots\}$.

Значення x_1=0,19 задовольняє співвідношенню:

$$0{,}10937 < 0{,}19 \le 0{,}34375, \text{ тобто}$$

$$\mathrm{p}_1 + \mathrm{p}_2 < \mathrm{x}_1 \le \mathrm{p}_1 + \mathrm{p}_2 + \mathrm{p}_3.$$

Тому, як випливає з графіка на рис. 3.8, значенням x_1= 0,19 відповідає значення y_1= 2.

Значення x_2= 0,94 задовольняє співвідношенню:

$$0{,}89063 < 0{,}94 \le 0{,}98438,$$

тобто $p_1 + p_2 + p_3 + p_4 + p_5 < x_2 \le p_1 + p_2 + p_3 + p_4 + p_5 + p_6.$

Таким чином, з графіка на рис. 3.8, значенням x_1 = 0,94 відповідає значення y_2 = 5.

Аналогічно міркуючи, показуємо, що значенню x_3 = 0,69 відповідає значення y_3 = 4, а значенню x_4 = 0,44 відповідає значення y_4 = 3.

У цілому базова послідовність $\{x_i\}$ ={0,19; 0,94; 0,69; 0,44; …} відповідає послідовності значень $\{y_i\}$ ={2, 5, 4, 3, …} дискретної випадкової величини *y*, що визначається біноміальним законом розподілу.

Для моделювання випадкових величин крім інтегрального закону розподілу використовується і диференціальний закон розподілу. Нехай випадкова величина η має диференціальний закон розподілу $f_\eta(y)$. Випадкова величина ξ, визначена як

$$\xi = \int_0^\eta f_\eta(y)dy\ ,$$

є рівномірно розподіленою в інтервалі (0,1).

Тоді для визначення значення y_i випадкової величини η, що має диференціальний закон розподілу $f_\eta(y)$, можна скористатися рівнянням

$$x_i = \int_0^{y_i} f_\eta(y)dy, \tag{3.6}$$

де x_i – деяке значення з базової послідовності $\{x_i\}$.

Наприклад, якщо необхідно отримати значення y_i випадкової величини η, що має показниковий закон розподілу

$$f_\eta(y) = \lambda e^{-\lambda y} \quad (y > 0),$$

то в цьому випадку рівняння (3.6) приймає вигляд:

$$x_i = \lambda \int_0^{y_i} e^{-\lambda y} dy. \tag{3.7}$$

Перетворюючи (3.7), отримуємо:

$$y_i = -\frac{1}{\lambda}\ln(1 - x_i), \tag{3.8}$$

де значення x_i з базової послідовності $\{x_i\}$. Беручи різні значення x із цієї послідовності і використовуючи співвідношення (3.8), формуємо послідовність $\{y_i\}$ випадкової величини η, що має показовий закон розподілу. Розглянутий метод отримання послідовності $\{y_i\}$ на основі заданого диференціального закону розподілу $f_\eta(y)$ і базової послідовності $\{x_i\}$ має обмежене використання через труднощі розв'язання рівнянь вигляду (3.6).

На практиці часто використовуються наближені методи отримання послідовностей значень випадкових величин, що відповідають заданому закону розподілу. Серед цих методів важливе значення мають універсальні методи. Їх універсальний характер полягає в тому, що вони не залежать від виду закону розподілу.

Відповідно до першого універсального методу, диференціальний закон розподілу $f_\eta(y)$ на інтервалі зміни значень випадкової величини (a, b) подається у вигляді кусково-постійної функції.

Для цього інтервал (a, b) розбивається на m під інтервалів (a_{k-1}, a_k), де $k \in \overline{1, m}$, $a_0 = a$, $a_m = b$, на кожному з яких $f_\eta(y)$ має постійне значення, що і показано на рис. 3.9.

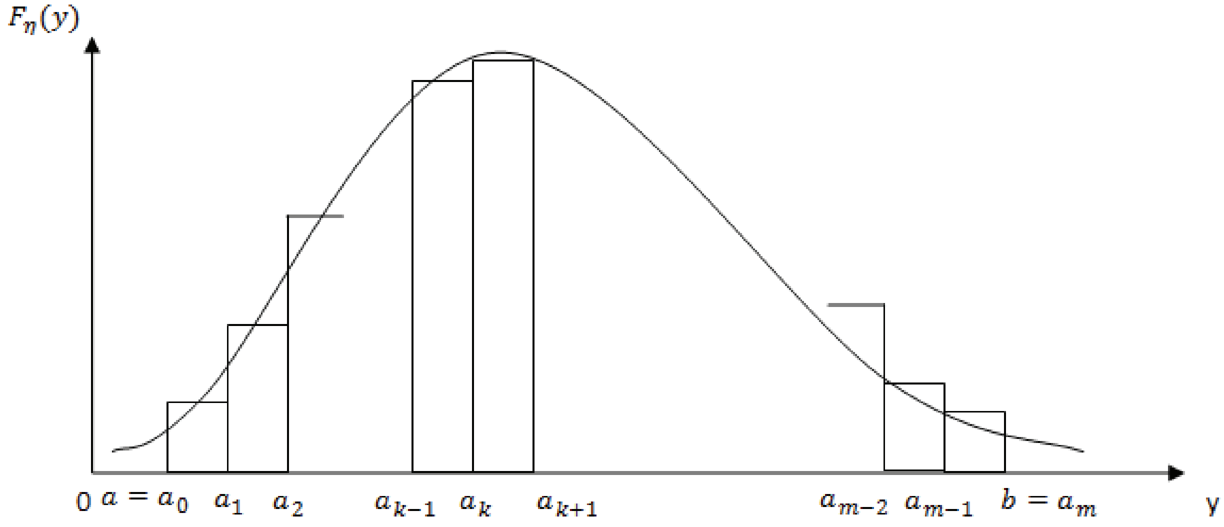


Рисунок 3.9 – Подання $f_\eta(y)$ у вигляді кусково-постійної функції на інтервалі (a, b)

Тоді випадкову величину η можна показати у вигляді

$$\eta = a_{k-1} + \eta_k,$$

де η_k – випадкова величина, рівномірно розподілена на інтервалі (a_{k-1}, a_k).

Визначення значення y_i, яке включається у формовану послідовність $\{y_i\}$, реалізується у два етапи. На першому з них визначається підінтервал (a_{k-1}, a_k), якому належить значення y_i, а на другому – саме значення y_i з цього підінтервалу.

Реалізація першого етапу ґрунтується на тому, що ймовірність попадання випадкової величини η на будь-який з підінтервалів (a_{k-1}, a_k) $k \in \overline{1,m}$ була постійною і не залежала від k. Це забезпечується тим, що

$$\forall k \in \overline{1,m} \quad \int_{a_{k-1}}^{a_k} f_\eta(y)dy = \frac{1}{m}. \tag{3.9}$$

Рівняння (3.9) являють собою рівняння з одним невідомим, послідовне розв’язання яких, починаючи з $k = 1$ $(a_0 = a)$, дозволяє отримати значення границь усіх m під інтервалів (a_{k-1}, a_k) $k \in \overline{1,m}$.

Для визначення, якому підінтервалу (a_{k-1}, a_k) належить значення y_i, використовується процедура жеребкування. З базової послідовності $\{x_i\}$ вибирається деяке значення x_i'.

Якщо $0 < x_i' \le \frac{1}{m}$, то $y_i \in (a_0, a_1)$.

Якщо $\frac{1}{m} < x_i' \le \frac{2}{m}$, то $y_i \in (a_1, a_2)$.

І так далі, якщо нарешті

$$\frac{m-1}{m} < x_i' \le 1, \text{ то } y_i \in (a_{m-1}, a_m),$$

де $a_m = b$.

У загальному вигляді, якщо $\frac{k-1}{m} < x_i' \le \frac{k}{m}$, то $y_i \in (a_{k-1}, a_k)$, $k \in \overline{1,m}$.

Для реалізації другого етапу з базової послідовності $\{x_i\}$ вибирається ще одне конкретне значення x_i''. Якщо на першому етапі було визначено інтервал (a_{k-1}, a_k), то значення y_i визначається на основі виразу

$$y_i = a_{k-1} + (a_k - a_{k-1})x_i''.$$

Отримане значення y_i включається у формовану послідовність $\{y_i\}$. Етапи 1, 2 повторюються стільки разів, скільки членів послідовності $\{y_i\}$ необхідно отримати.

Другим універсальним наближеним методом є метод виключення, відповідно до якого вся крива $f_\eta(y)$ охоплюється прямокутником зі сторонами (*a, b*) по вісі *y* і $(0, f^*)$, де $f^* = \max\limits_{y \in (a,b)} f_\eta(y)$, по вісі значень $f_\eta(y)$ так, як показано на рис. 3.10.

Далі реалізується наступний алгоритм.

Крок 1. З базової послідовності $\{x_i\}$ вибираються два значення x_i' і x_i'', які зв'язуються з довжинами сторін (*a, b*) і $(0, f^*)$ так, що

$$\overline{x}_i' = a + (b-a)x_i',$$

$$\overline{x}_i'' = 0 + (f^* - 0)x_i'' = f^* x_i''.$$

Крок 2. Якщо виявиться, що $\overline{x}_i'' \le f_\eta(\overline{x}_i')$, тобто точка з координатами $(\overline{x}_i', \overline{x}_i'')$ виявляється під кривою $f_\eta(y)$, як показано на рис. 3.10, то вважають $y_i = \overline{x}_i'$, і знайдене значення включають у послідовність $\{y_i\}$, що формується.

В іншому випадку, коли $\overline{x}_i'' > f_\eta(\overline{x}_i')$, тобто точка з координатами $(\overline{x}_i', \overline{x}_i'')$ виявляється над кривою $f_\eta(y)$, то значення y_i вважається невизначеним, а точка $(\overline{x}_i', \overline{x}_i'')$ виключається з розгляду.

Повертаємося на початок першого кроку.

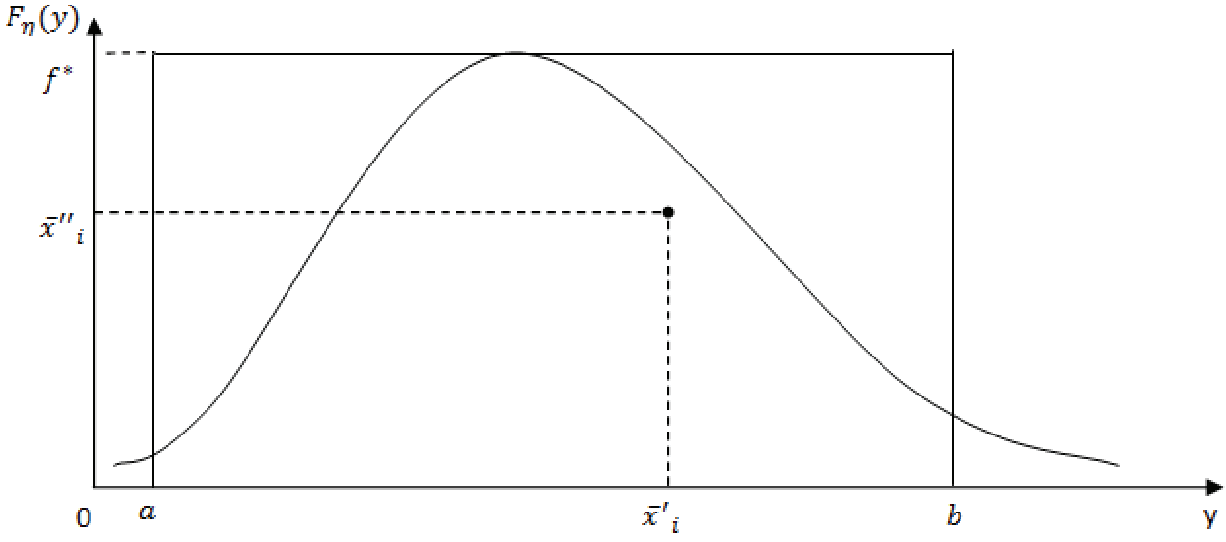


Рисунок 3.10 – Апроксимація $f_\eta(y)$ прямокутником зі сторонами (*a, b*) та $\left(0, f^*\right)$

Дії зазначених кроків повторюються до тих пір, поки не буде сформована послідовність значень $\{y_i\}$ необхідної довжини.

Число реалізацій кроків алгоритму істотно залежить від виду функції $f_\eta(y)$. Очевидно, що для функції, показаної на рис. 3.11, знадобиться більше реалізацій для формування послідовності заданої довжини $\{y_i\}$, ніж для функції $f_\eta(y)$ на рис. 3.10, оскільки площа під кривою на рис. 3.10 значно більше площі під кривою на рис. 3.11 при незмінній площі прямокутника, що охоплює ці криві.

Метод виключення може бути використаний для моделювання випадкового вектора, тобто вектора, компоненти якого являють собою систему випадкових величин із заданим спільним законом розподілу.

Наприклад, нехай розглядається випадковий вектор $\eta = (\eta_1, \eta_2)$, компоненти якого випадкові величини η_1, η_2 із заданим спільним диференціальним законом розподілу (спільною густиною ймовірності) $f_{\eta_1,\eta_2}(y_1, y_2)$.

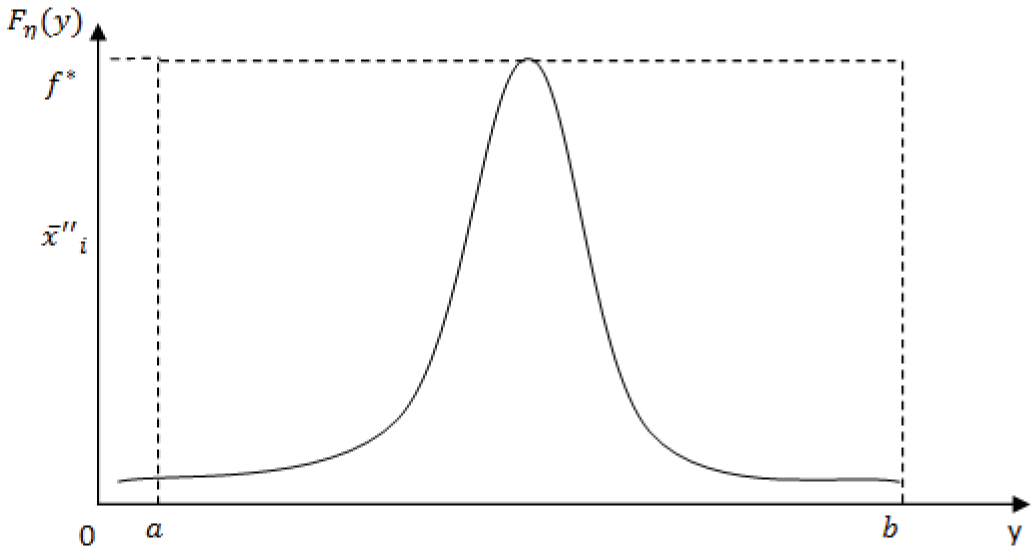


Рисунок 3.11 – Вигляд деякої функції $f_\eta(y)$

Моделювання вектора η полягає в отриманні на основі базової послідовності $\{x_i\}$ послідовності $\{y_i\} = \{(y_{1i}, y_{2i})\}$ заданої довжини. Моделювання реалізується виконанням наступного алгоритму.

Крок 1. З базової послідовності $\{x_i\}$ вибираються три значення x_i, x_i', x_i'', які перетворюються в значення

$$\bar{x}_i = a_1 + (b_1 - a_1)x_i;$$

$$\bar{x}_i' = a_2 + (b_2 - a_2)x_i';$$

$$\bar{x}_i'' = f^* x_i'',$$

де $f^* = \max\limits_{y_1 y_2} f_{\eta_1,\eta_2}(y_1, y_2)$, $y_1 = (a_1, b_1)$, $y_2 = (a_2, b_2)$.

Крок 2. Якщо виконується умова $\bar{x}_i'' \le f_{\eta_1,\eta_2}(\bar{x}_i, \bar{x}_i')$, що означає попадання випадкової точки з координатами $(\bar{x}_i, \bar{x}_i', \bar{x}_i'')$ під поверхню $f_{\eta_1,\eta_2}(y_1, y_2)$, то $y_{1i} = \bar{x}_i$, $y_{2i} = \bar{x}_i'$ і одержувана реалізація вектора $y_i = (y_{1i}, y_{2i})$ включається у послідовність $\{y_i\} = \{(y_{1i}, y_{2i})\}$, що формується.

Повертаємося на початок першого кроку і повторюємо описані дії доти, поки не буде сформована послідовність $\{y_i\}$ заданої довжини.

Обчислювальна процедура отримання реалізацій випадкових векторів великої розмірності методом виключення настільки громіздка, що на практиці вона рідко використовується. У цьому випадку використовуються різні спеціальні методи, що базуються на розрахунку оцінок числових характеристик.

Висновки

Статистичне моделювання широко застосовується для вивчення стохастичних об'єктів, описуваних за допомогою таких математичних схем, як імовірнісні автомати та системи масового обслуговування. У даній главі розглядаються основні напрями використання методу статистичного моделювання. Описано різні методи генерації псевдовипадкових чисел, які використовуються в статистичному моделюванні. Розглянуто моделювання випадкових впливів.

Завдання

1. Складіть формулу стандартного методу моделювання для випадкової величини з густиною розподілу $ce^{-\sigma x}$; $0 \le x \le a$; $\sigma > 0$.
2. Складіть формулу стандартного методу моделювання для випадкової величини з густиною розподілу $c(x+1)^{0.75}$; $0 \le x \le 1$.
3. Складіть формулу стандартного методу моделювання для випадкової величини з густиною розподілу $c|\sin x|$; $-\frac{\pi}{2} \le x \le \frac{\pi}{2}$.
4. Складіть формулу стандартного методу моделювання для випадкової величини з густиною розподілу $\sum_{n=0}^{\infty} \frac{(n+1)(\lambda x)^n}{n!} e^{-\lambda}$; $0 \le x \le 1$; $\lambda > 0$.
5. Розробіть алгоритм моделювання для випадкової величини з густиною розподілу $\sum_{n=0}^{\infty} (n+1) p(x(1-p))^n$; $0 \le x \le 1$; $0 < p < 1$.
6. Розробіть алгоритм моделювання для випадкової величини з густиною розподілу $c\left(x^{-2} + x^2 e^{-x^3}\right)$; $x \ge 1$.
7. Розробіть алгоритм моделювання для випадкової величини з густиною розподілу $ce^{-ax}\left(1 + e^{-bx}\right)$; $x \ge 0$; $a, b > 0$.
8. Складіть моделюючу формулу для випадкової величини з густиною розподілу $ce^{-(x-1)^2}$; $x \ge 1$.
9. Складіть моделюючу формулу для випадкової величини з густиною розподілу $cx^{10} e^{-\sigma x}$; $x \ge 0$; $\sigma > 0$.
10. Складіть моделюючу формулу для випадкової величини з густиною розподілу $c(1-x^3)$; $0 \le x \le 1$.
11. Складіть моделюючу формулу для випадкової величини з густиною розподілу $c(\ln x)^4$; $0 \le x \le 1$.

12. Розробіть алгоритм моделювання для випадкової величини з густиною розподілу $cx(1-x)^2$; $0 \le x \le 1$.
13. Розробіть алгоритм моделювання для випадкової величини з густиною розподілу $c(1-x)^3$; $0 \le x \le 1$.

Контрольні питання

1. Складовою частиною яких математичних схем виступає метод статистичного моделювання?
2. Які відомі основні напрямки використання методу статистичного моделювання?
3. Які існують способи генерації випадкових чисел?
4. У чому полягають недоліки апаратного способу генерації випадкових чисел?
5. У чому полягають конгруентні процедури генерації послідовності псевдовипадкових величин?
6. Опишіть мультиплікативний метод отримання псевдовипадкових чисел.
7. Як реалізується перевірка рівномірності, стохастичності і незалежності одержуваних псевдовипадкових квазірівномірно розподілених чисел?
8. Як моделюється окрема випадкова подія?
9. Як моделюється складна подія, залежна від двох і більше простих подій?
10. Як моделюється однорідний марківський ланцюг?
11. У чому полягає метод зворотної функції для моделювання випадкових впливів у вигляді випадкових дискретних і неперервних величин?
12. Які існують універсальні методи отримання послідовностей значень випадкових величин?

4. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ

4.1. Сутність імітаційного моделювання та основні випадки його використання

Реальні складні системи, як правило, досліджуються за допомогою двох видів математичних моделей: аналітичної та імітаційної, а також на основі їх спільного використання (комбіноване моделювання).

В аналітичних моделях функціонування складної системи представляється системою функціональних співвідношень, які можуть доповнюватися різноманітними логічними умовами. Часто така система є системою диференціальних, скінченно-різницевих та інших видів рівнянь.

Найбільш повні дослідження за допомогою аналітичного моделювання вдається провести, коли отримують явні залежності між величинами, що характеризують досліджувану якість реальної системи, і параметрами системи. Для отримання таких залежностей доводиться спрощувати явища, які спостерігаються в реальних системах.

Коли явища в системі настільки складні, що їх спрощення стає занадто грубим наближенням до дійсності, слід відмовитися від аналітичного моделювання і використовувати для дослідження подібних систем імітаційне моделювання.

Імітаційне моделювання не передбачає тих спрощень явищ у реальній системі, які необхідні для реалізації аналітичного моделювання. В імітаційній моделі функціонування складної системи подається набором алгоритмів. Ці алгоритми на основі фактичних значень параметрів і відомостей про початковий стан системи дозволяють відтворити функціонування системи в кожній конкретній ситуації.

Найбільш часто імітаційне моделювання використовується в наступних випадках:

- коли імітаційне моделювання виявляється єдиною можливістю дослідження складної системи через труднощі спостереження явищ у реальних умовах;

- коли необхідно дослідити процеси у складній системі шляхом їх штучного уповільнення або прискорення;
- при підготовці фахівців для роботи з новою технікою, коли імітаційна модель є засобом для придбання навичок експлуатації нової техніки;
- коли вивчаються нові ситуації у складних системах, про які мало що відомо або невідомо нічого, і тому імітаційна модель є єдиним засобом для перевірки стратегій і правил поведінки в нових ситуаціях;
- коли особливе значення має послідовність подій у складній системі й імітаційна модель використовується для виявлення можливих «вузьких місць» та інших труднощів у функціонуванні системи, що вивчається, при введенні до неї нових елементів.

До основних достоїнств імітаційного моделювання відносять:

- можливість опису явищ і процесів у складних системах на високому рівні деталізації;
- відсутність обмежень на вигляд залежностей функціональних відносин, використовуваних для опису функціонування складних систем (нелінійність, нестаціонарність, стохастичність і т.д.);
- можливість дослідження динаміки взаємодії елементів і компонентів системи у просторі параметрів і в часі.

Однак є й недоліки імітаційного моделювання:

- розроблення імітаційної моделі складної системи, як правило, вимагає досить великих витрат ресурсів і часу;
- імітаційна модель як будь-яка модель не є точною, але при цьому ступінь неточності оцінити важко або неможливо.

Термін «імітаційне моделювання» означає, що за допомогою таких математичних моделей результат не обчислюється. Результат, який цікавить дослідника, за допомогою імітаційного моделювання отримують на основі обробки експерименту (імітації) на моделі при заданих вихідних даних. Експеримент при цьому полягає в реалізації алгоритму або множини алгоритмів, що відображають функціонування досліджуваної системи. Імітація являє собою чисельний метод проведення на комп’ютері експериментів з математичними моделями, що описують поведінку складної системи протягом заданого або того періоду часу, що формується.

4.2. Модельний час і способи його реалізації

Побудова імітаційної моделі складної системи передбачає структуроване представлення системи, в якій виділяються компоненти, тобто $S' = \{K_i \mid i \in \overline{1,n}\}$, де K_i – i-а компонента системи.

Функціонування компоненти K_i являє собою послідовність функціональних дій $ФD_{ij}$ $(j \in \overline{1,m})$. Інакше кажучи, $K_i = \{ФD_{ij} \mid j \in \overline{1,m_i}\}$, де

$$ФD_{i1} \prec ФD_{i2} \prec \ldots \prec ФD_{ij} \prec \ldots \prec ФD_{im_i}. \quad (4.1)$$

В результаті виконання кожної функціональної дії $ФD_{ij}$ в складній системі відбувається подія C_{ij}.

Будь-яка $ФD_{ij}$ виконується на деякому часовому інтервалі τ_{ij}. Для характеристики функціонування K_i у часі вводиться локальний час t_i.

У складній системі всі локальні часи $t_i \left(i \in \overline{1,n}\right)$ змінюються одночасно, однак характер цих змін різний, і він визначається послідовністю часових інтервалів τ_{ij} відповідній послідовності функціональних дій $ФD_{ij}$.

При побудові імітаційної моделі складної системи $ФD_{ij}$ апроксимуються спрощеними функціональними діями $ФD'_{ij}$. Ступінь спрощення визначає рівень деталізації імітаційної моделі. Розбіжності між $ФD_{ij}$ і $ФD'_{ij}$ утворюють неточність імітаційної моделі, оцінку якої, як уже зазначалося, досить важко отримати.

В імітаційній моделі реальна функціональна дія $ФD_{ij}$ подана парою $< ФD'_{ij}, \tau_{ij} >$, яка виконується в наступному порядку.

Спочатку при фіксованому значенні локального часу t_i виконується спрощена дія $ФD'_{ij}$, а потім змінюється t_i на величину τ_{ij}. Після зміни t_i на величину τ_{ij} ініціюється виникнення події C_{ij}, яка підтверджує факт завершення виконання дії $ФD_{ij}$ в реальній складній системі.

Впорядковане згідно з послідовністю (4.1) виконання всіх функціональних дій зазначеним чином в імітаційній моделі визначає порядок реалізації імітаційного моделювання кожної компоненти K_i складної системи S'. Цей порядок схематично показано на рис. 4.1.

Згідно зі схемою на рис. 4.1 при $t_i = 0$ спочатку виконується перша спрощена функціональна дія $ФD'_{i1}$. Потім значення $t_i = 0$ змінюється на величину τ_{i1} і стає рівним $t_i = \tau_{i1}$. У цей момент часу виникає подія C_{i1}, що підтверджує факт виконання реальної дії ΦD_{i1}, яка умовно показано на рис. 4.1 пунктиром.

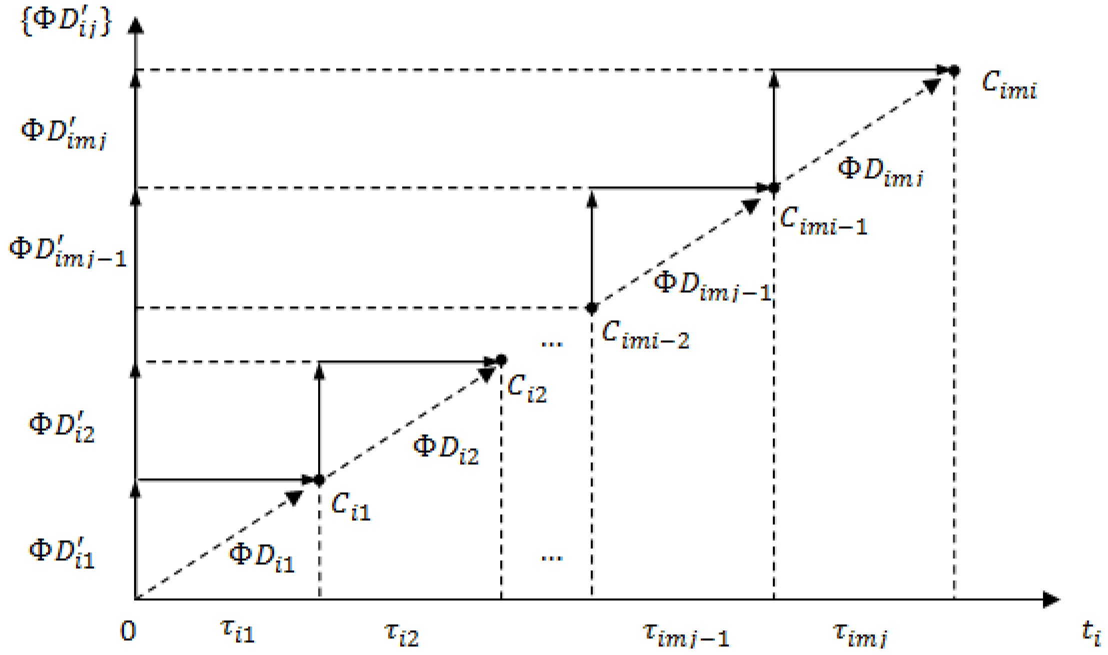


Рисунок 4.1 – Схема реалізації імітаційної моделі компоненти K_i

Далі в момент $t_i = \tau_{i1}$ виконується спрощена дія $\Phi D'_{i2}$. Після її виконання значення $t_i = \tau_{i1}$ змінюється на величину τ_{i2} і стає рівним $t_i = \tau_{i1} + \tau_{i1}$. У цей момент ініціюється подія C_{i2}, що підтверджує виконання реальної дії ΦD_{i2}, показаної на рис. 4.1 пунктиром, і т.д.

В імітаційній моделі кожна спрощена дія $\Phi D'_{i1}$ реалізується відповідним алгоритмом АЛ_{ij}. Таким чином, реальна функціональна дія ΦD_{ij} в імітаційній моделі, подана алгоритмом АЛ_{ij} і оператором Mt_{ij}, змінює локальну тимчасову координату t_i на величину τ_{ij}.

Пара $< \text{АЛ}_{ij}, \tau_{ij} >$ називається активністю імітаційної моделі і позначається як АК_{ij}. Реалізація активності $\text{АК}_{ij} =< \text{АЛ}_{ij}, \tau_{ij} >$ призводить до появи в імітаційній моделі події C_{ij}.

У цілому імітаційну модель складної системи ІМ можна розглядати як множину активностей, тобто

$$\text{ІМ} = \left\{ \text{АК}_{ij} \mid i \in \overline{1,n},\ j \in \overline{1,m_i} \right\},$$

з урахуванням порядку реалізації активностей, що задається співвідношеннями (4.1) для кожної компоненти K_i.

Компоненти складної системи $K_i\,(i \in \overline{1,n})$ функціонують одночасно або, інакше кажучи, паралельно. Однак у сучасних комп'ютерах у кожен момент часу реалізується алгоритм або активність, яка належить тільки до однієї з компонент. Тому виникає проблема – як за допомогою комп'ютера, що послідовно обробляє інформацію, відобразити паралельну роботу декількох компонент складної системи.

Зазначена проблема вирішується шляхом введення деякої глобальної змінної t_0, яку називають модельним або системним часом. За допомогою цієї змінної організується синхронізація подій C_{ij} у моделі або, інакше кажучи, виконання активностей АК_{ij} різних компонент $K_i\,(i \in \overline{1,n})$ складної системи.

Таким чином, при реалізації імітаційної моделі використовуються три подання часу: реальний час t_r, модельний або системний час t_0, машинний час t_k, який виражає витрати ресурсу часу комп'ютера на виконання імітації.

По суті модельний час t_0 дозволяє реалізувати квазіпаралельну роботу компонентів складної системи, тобто фіксуються одночасно події, що виникають у різних компонентах системи, а потім вони послідовно обслуговуються.

Зміни локальних координат реального часу t_i на основі модельного часу t_0 відбуваються відповідно до наступного прикладу. Нехай є дві компоненти K_1 та K_2, кожна з яких функціонує у відповідному локальному часі t_1 та t_2. І нехай в деякий момент модельного часу t'_0 в компонентах K_1 та K_2 одночасно відбуваються події C_{1j_1} і C_{2j_2}. Якщо через t_{1j_1} та t_{2j_2} позначити моменти настання цих подій, то очевидно $t'_0 = t_{1j_1} = t_{2j_2}$, що і показано на рис. 4.2.

Відповідно до встановленого порядку виконання імітаційної моделі, спочатку послідовно виконуються алгоритми АЛ_{1j_1+1}, АЛ_{2j_2+1} (або навпаки) при фіксованому значенні як локальних координат часу t_i, так і модельного часу t_0, а потім змінюються локальні координати t_i на величину тривалості виконання відповідної дії τ_{ij}. Таким чином, нові значення локальних координат часу визначаються на основі t_0 як

$$t_{1j_1+1} = t'_0 + \tau_{1j_1+1};\quad t_{2j_2+1} = t'_0 + \tau_{2j_2+1}.$$

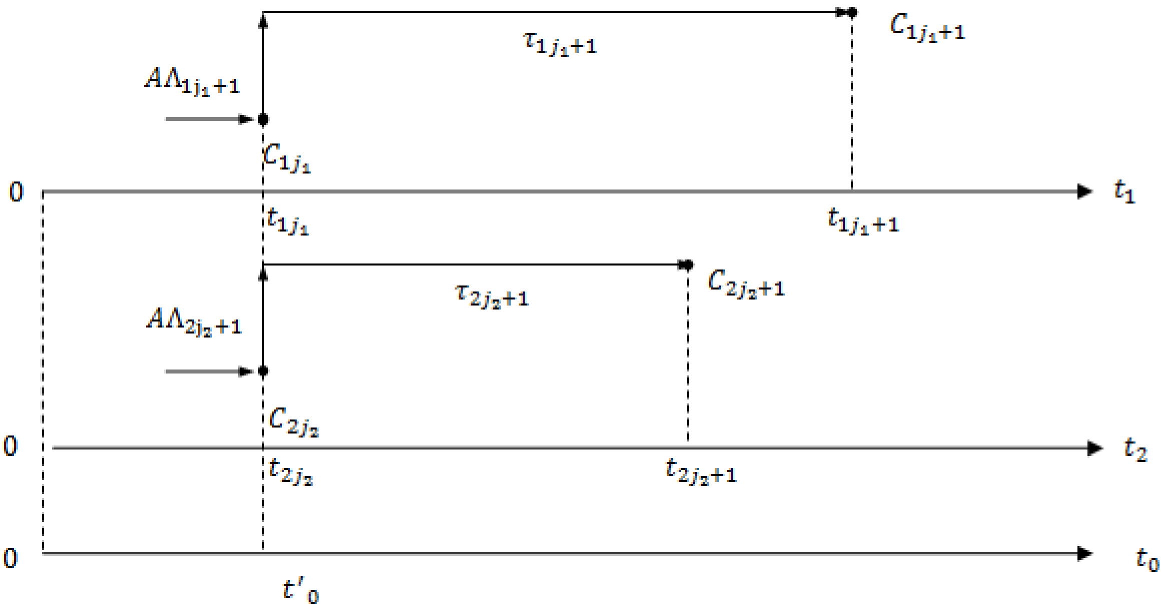


Рисунок 4.2 – Схема зміни t_i на основі t_0

Отримані нові значення t_{1j_1+1}, t_{2j_2+1} запам'ятовуються і використовуються надалі для наступних активацій компонент $K_i\ (i \in \overline{1,2})$ в імітаційній моделі. При цьому під активацією розуміється початок виконання наступної активності. У розглянутому прикладі, очевидно, це будуть активності АК_{1j_1+1} та АК_{2j_2+1}. Активація зводиться до виконання відповідного алгоритму і зміни локальної тимчасової координати.

Після зміни всіх локальних часових координат $t_i\ (i \in \overline{1,n})$ виникає проблема визначення нового значення модельного часу t_0. Найбільш часто використовуються два способи зміни модельного часу t_0, кожен з яких має кілька назв. Перший з них називають методом фіксованого інтервалу, або фіксованого кроку, або принципом Δt. Другий метод називають методом змінного інтервалу, або кроку до наступної події, або принципом δz.

Сутність цих двох методів розглянемо на наступному прикладі. Нехай в системі три компоненти, тобто $S' = \{K_i \mid i \in \overline{1,3}\}$, процеси функціонування яких зводяться до виконання функціональних дій. У першій компоненті виконуються чотири функціональні дії, тобто

$$K_1 = \{\Phi D_{1j} \mid j \in \overline{1,4}\},$$

$$\text{де } \Phi D_{11} \prec \Phi D_{12} \prec \Phi D_{13} \prec \Phi D_{14},$$

а у другій і третій компонентах виконуються по три функціональні дії, тобто

$$K_2 = \{\Phi D_{2j} \mid j \in \overline{1,3}\},$$

$$\Phi D_{21} \prec \Phi D_{22} \prec \Phi D_{23},$$

$$K_3 = \{\Phi D_{3j} \mid j \in \overline{1,3}\},$$

$$\Phi D_{31} \prec \Phi D_{32} \prec \Phi D_{33}.$$

Відомі тривалості виконання всіх функціональних дій $\{\tau_{ij}\}$.

Досить загальним видом імітаційної моделі зазначеної складної системи є множина апроксимуючих функціональні дії активностей, які виконуються в порядку, аналогічному порядку виконання функціональних дій у кожній компоненті, тобто

$$\text{IM} = \{\text{AK}_{ij_i} \mid i \in \overline{1,3},\ j_1 \in \overline{1,4},\ j_2 \in \overline{1,3},\ j_3 \in \overline{1,3}\}$$

$$\text{AK}_{11} \prec \text{AK}_{12} \prec \text{AK}_{13} \prec \text{AK}_{14};$$

$$\text{AK}_{21} \prec \text{AK}_{22} \prec \text{AK}_{23};$$

$$\text{AK}_{31} \prec \text{AK}_{32} \prec \text{AK}_{33}.$$

Кожне виконання активностей AK_{ij} призводить до появи в моделі відповідної події C_{ij}.

З урахуванням порядку виконання імітаційної моделі і значень тривалостей виконання дій $\{\tau_{ij}\}$ схема зміни локальних координат часу t_i і модельного часу t_0 подана на рис. 4.3.

При цьому t'_0 характеризує зміну модельного часу за першим способом, а t''_0 – за другим.

При імітації за першим способом модельний час t'_0 змінюється щоразу на величину Δt у моменти модельного часу $0, \Delta t, 2\Delta t$ і т.д. В імітаційній моделі відбуваються події S_0, S_1, S_2 і т.п., які можна розглядати як своєрідні сигнали керуючої програмі моделювання (КПМ) на обслуговування подій C_{ij}, потрапляють у черговий інтервал Δt.

Наприклад, коли $t'_0 = 0$, настає подія S_0 і КПМ обслуговує всі події C_{ij} на інтервалі $(0, \Delta t)$ або (S_0, S_1). З рис. 4.3 випливає, що в інтервал (S_0, S_1) потрапляють події C_{11} і C_{31}. Вважається, що ці події відбуваються одночасно і момент їх настання приписується до правого кінця інтервалу, на який вони потрапили, тобто до моменту $t'_0 = \Delta t$. КПМ послідовно обслуговує події C_{11} і C_{31}, що зводиться до ініціалізації відповідних активностей АК$_{11}$ і АК$_{31}$, тобто до виконання алгоритмів АЛ$_{11}$, АЛ$_{31}$ і зміни часу на величини τ_{11}, τ_{31}.

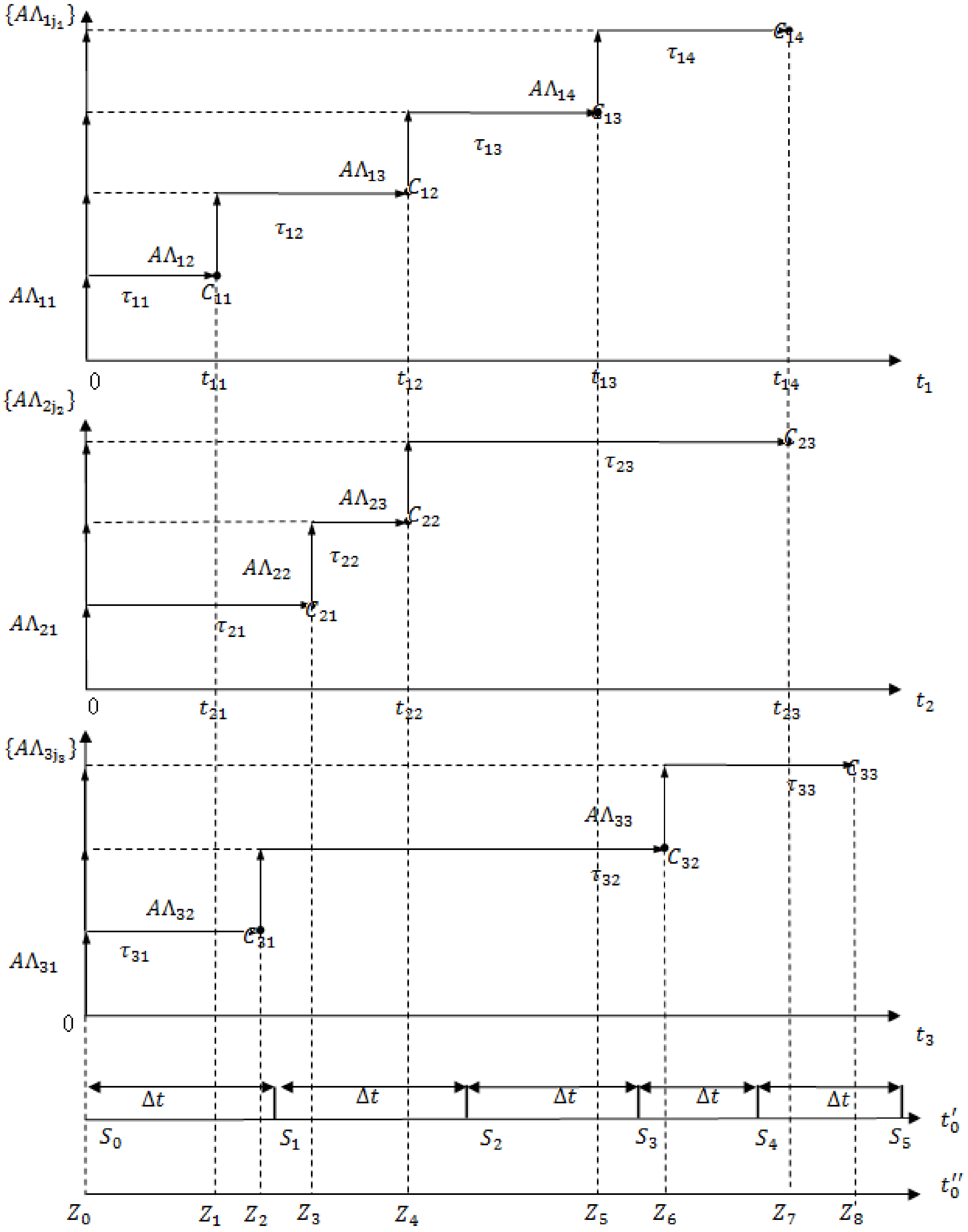


Рисунок 4.3 – Схема виконання імітаційної моделі з урахуванням зміни локальних координат часу t_i і модельного часу двома способами t'_0, t''_0

У наступний момент модельного часу $t'_0 = \Delta t$ настає подія S_1 і КПМ обслуговує подію C_{ij} на інтервалі $(\Delta t, 2\Delta t)$ або (S_1, S_2). Це події C_{21}, C_{12}, C_{22}, що відбуваються, як вважають, одночасно в момент $t'_0 = 2\Delta t$. КПМ послідовно обслуговує події, що зводиться до ініціалізації активностей АК_{21}, АК_{12}, АК_{22} і т.д.

Таким чином, у цілому механізм зміни модельного часу за першим способом зводиться до збільшення на заздалегідь обрану величину Δt з подальшим обслуговуванням КПМ подій C_{ij}, які потрапляють на черговий інтервал Δt. При цьому всі ці події належать до правої межі інтервалу модельного часу.

Очевидно, що точність моделювання залежить від Δt. Вона зростає зі зменшенням Δt, але при цьому зростає і число звернень до КПМ, що призводить до збільшення витрати ресурсу часу комп'ютера t_k на імітацію.

Відповідно до другого способу зміни модельного часу, t''_0 приймає значення Z_l, відповідають моментам t_{ij}, в які відбуваються події C_{ij}. Якщо виявиться, що деякому моменту t''_0 дорівнюють кілька значень t_{ij}, тобто в момент t''_0 відбувається одночасно кілька подій C_{ij}, то ці події послідовно обслуговуються КПМ без зміни модельного часу. Наприклад, згідно з рис. 4.3, момент $t''_0 = z_4 = t_{12} = t_{22}$. Це означає, що в момент $t''_0 = z_4$ одночасно відбуваються дві події – C_{12} і C_{22}. КПМ послідовно обслуговує ці події, що зводиться до ініціалізації активностей АК_{12} і АК_{22}, тобто до виконання алгоритмів АЛ_{12}, АЛ_{22} і зміні часів t_1, t_2 на величини τ_{12}, τ_{22}. Модельний час t''_0 приймає значення найближчого t_{ij} на вісях локальних координат t_i.

Незалежно від способу зміни t_0 механізм зміни модельного часу включає наступні дії:

- визначаються події, які необхідно обслужити при даному модельному часі t_0;
- обслуговуються події шляхом ініціалізації відповідних активностей;
- визначається наступне значення модельного часу t_0 і проводиться його коректування;
- перевіряється умова закінчення моделювання, яка найчастіше полягає в порівнянні поточного часу з наперед заданим або в настанні заданих подій.

Усі зазначені дії виконуються КПМ. Обслуговування подій C_{ij} вимагає витрат ресурсу часу комп'ютера t_k. Очевидно, що чим частіше обслуговуються події, тим більше значення t_k. З цієї точки зору спосіб фіксованого кроку зміни модельного часу t_0 є більш економічним, оскільки відбувається групове обслуговування всіх подій C_{ij}, які потрапляють всередину чергового кроку Δt зміни модельного часу.

На практиці способу фіксованого кроку перевага віддається у двох випадках:

- коли події C_{ij} розподіляються рівномірно на всьому інтервалі моделювання і при цьому вдається вибрати Δt так, що забезпечується мінімальна похибка імітації;
- коли події C_{ij} з'являються групами і вдається вибрати значення Δt , яке охоплює всі події групи.

У всіх інших випадках кращим є спосіб кроку до наступної події як більш точний і економний.

4.3. Структурна схема імітаційної моделі і технологічні етапи процесу побудови імітаційної моделі

Можна виділити чотири структури моделювання, визнані спільнотою імітаційного моделювання. До них належать метод процесної взаємодії, метод планування подій, метод сканування дій і трифазний метод.

Інтуїтивно найбільш зрозумілою імітаційної структурою є метод процесної взаємодії (рис. 4.4). У цьому методі програма симулює потік певного об'єкта (наприклад, load – навантаження) усередині системи. Об'єкт рухається по системі, поки він не входить в яку-небудь діяльність або не виходить із системи. Коли рух об'єкта припиняється, час просувається до моменту наступного об'єкта. Цей потік, або рух описує послідовність станів, в яких об'єкт може знаходитися в системі. Наприклад, у моделі пральні клієнт може увійти в систему, чекати вільну пральну машину, прати одяг у пральній машині, очікувати вільної корзини для білизни, розвантажувати пральну машину, переносити одяг у кошику для сушіння, завантажувати одяг для сушіння, сушити одяг, розвантажувати сушку, і нарешті, залишати сушку. Кожний стан і подія при цьому моделюється.

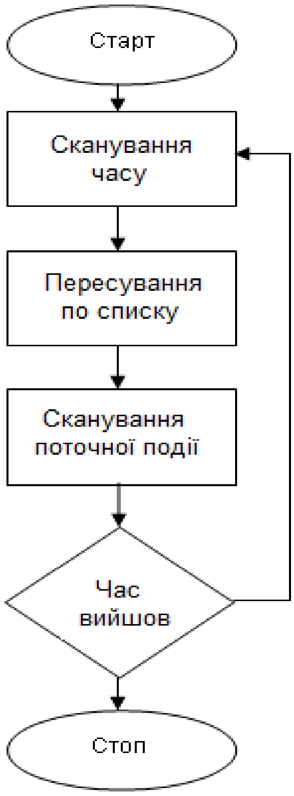


Рисунок 4.4 – Виконання методу процесної взаємодії

Метод потоку транзакцій є спрощеною версією методу процесної взаємодії. Вперше метод потоку транзакцій був реалізований засобом імітаційного моделювання GPSS у 1962 році. У цьому підході модель складається з сутностей (одиниць трафіку), ресурсів (елементів, які обслуговують сутності) і контрольних елементів (елементів, які визначають стан сутностей і ресурси). Часто на цьому підході ґрунтуються дискретні імітатори, створені для проведення імітаційного моделювання деталізованих процесів, наприклад, роботи центрів телефонного зв’язку, виробничих процесів на підприємстві і т.д.

Наступною структурою імітаційного моделювання є метод планування подій. Ключовим моментом у методі планування подій є просування часу до моменту, коли трапиться якась наступна подія (рис. 4.5). Це означає, що коли одна подія закінчується, час просувається до моменту наступної події. Зазвичай подія вивільняє який-небудь ресурс. Подія перерозподіляє доступні об’єкти чи сутності шляхом упорядкування дій, в яких вони можуть приймати участь в даний момент. Час просувається до наступної упорядкованої події (зазвичай закінчення дії). Потім усі дії проглядаються на предмет того, чи може якась початися в даний момент. Метод планування подій має наступну перевагу. Даний підхід не вимагає підтримки ніякої спеціалізованої програмної мови моделювання або операційної системи. Недоліком методу планування подій є те, що опис системи як набору подій не має значення для розуміння потоку процесу.

Третя структура імітаційного моделювання є методом сканування дій, також відомим як двофазний метод (рис. 4.6). Метод сканування дій надає програму імітації, що складається з незалежних модулів, які чекають того, щоб бути виконаними. Сканування відбувається у фіксовані проміжки часу, в які перевіряється, чи відбувається подія в цей момент. Якщо подія відбувається, стан системи змінюється.

Трьохфазовий метод являє четверту структуру імітаційного моделювання (рис. 4.7). У даному підході час просувається до тих пір, поки не настає зміна стану системи або наступна подія. Перша фаза – це просування часу. У другій фазі вивільняються ресурси, які повинні завершити свої дії до цього часу. У третій фазі починаються дії, що мають глобальну картину доступності ресурсів.

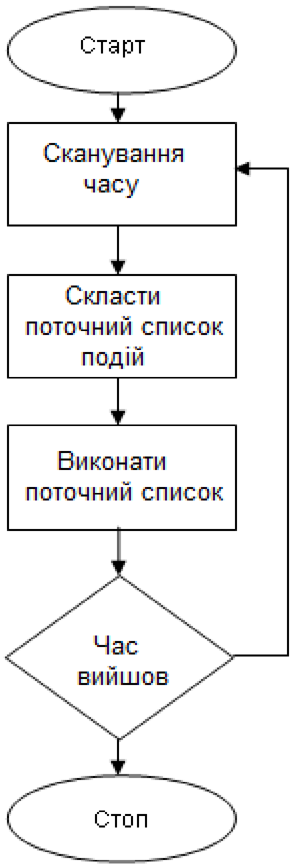


Рисунок 4.5 – Виконання методу планування подій

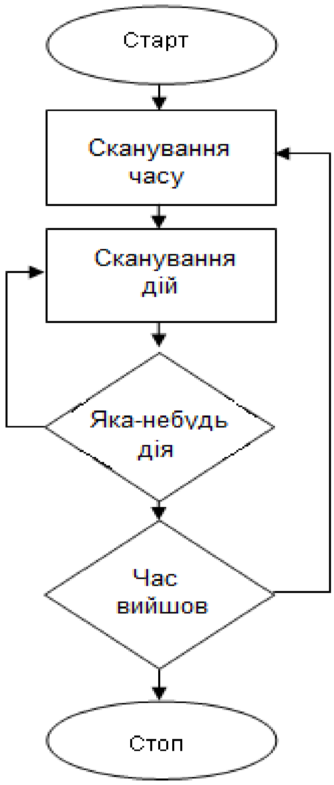


Рисунок 4.6 – Виконання методу сканування дій

Можна виділити деякі переваги всіх описаних підходів. Метод процесної взаємодії дозволяє уникати непродуктивних, повільних програмних реалізацій. Також, використовуючи цей підхід, можна не думати про всі можливі логічні наслідки події. Однак реалізація швидких програмних розв'язань у цьому випадку вимагає додаткових навичок.

Метод планування подій простіше за двох- і трьохфазовий методи. Це дозволяє програмі працювати швидше, оскільки сканування умовних подій тут відсутнє.

Метод сканування дій, у свою чергу, простіший за трьохфазовий метод. Однак при цьому метод не дозволяє реалізувати досить швидке програмне рішення.

Одним із структурних подань імітаційної моделі є кортеж, що включає множину активностей $\{\text{АК}_{ij_i} \mid i \in \overline{1,n},\ j \in \overline{1,m_i}\}$ і КПМ, забезпечує взаємодію активностей між собою в ході імітації та виконання організуючих функцій, тобто

$$\text{ІМ} = \left\langle \{\text{АК}_{ij} \mid i \in \overline{1,n},\ j \in \overline{1,m_i}\}, \text{КПМ} \right\rangle.$$

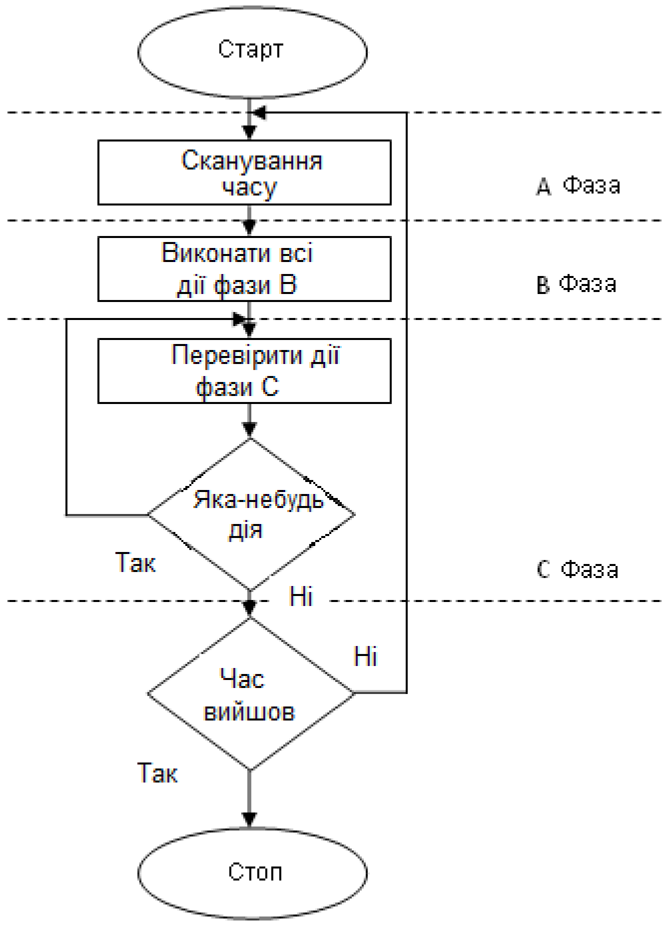


Рисунок 4.7 – Виконання трьохфазового методу

Організуючі функції КПМ полягають:

- у модифікації модельного часу t_0;
- у виконанні операторів Mt_{ij}, змінюють локальні координати t_i на величини τ_{ij};
- у запуску на виконання алгоритмів $АЛ_{ij}$;
- у перевірці умов завершення імітації.

У КПМ виділяються три основні підпрограми (ПП): ПП початку імітації, ПП збирання статистики, ПП закінчення імітації.

ПП початку імітації призначається для завдання вхідних умов і початкових значень параметрів моделі.

У ПП збирання статистики фіксуються статистики моделювання, що характеризують поведінку компонент $K_i\,(i \in \overline{1,n})$ складної системи в різних режимах її функціонування.

У ПП закінчення імітації перевіряється умова завершення імітації, і якщо вона виконується, то реалізується процес оброблення зібраних статистик моделювання, що дозволяє отримати оцінки характеристик поведінки досліджуваної системи і видати їх досліднику у необхідній формі.

У структурі імітаційної моделі виділяються дві частини. Одна, подана множиною активностей $\{АК_{ij} \mid i \in \overline{1,n},\ j \in \overline{1,m_i}\}$, визначається сутністю модельованих компонент $K_i\ (i \in \overline{1,n})$ складної системи, і тому має змінний характер. Інша, у вигляді КПМ, має досить фіксований набір елементів і постійне число виконуваних функцій.

Взаємодія зазначених частин у структурі імітаційної моделі показана на рис. 4.8.

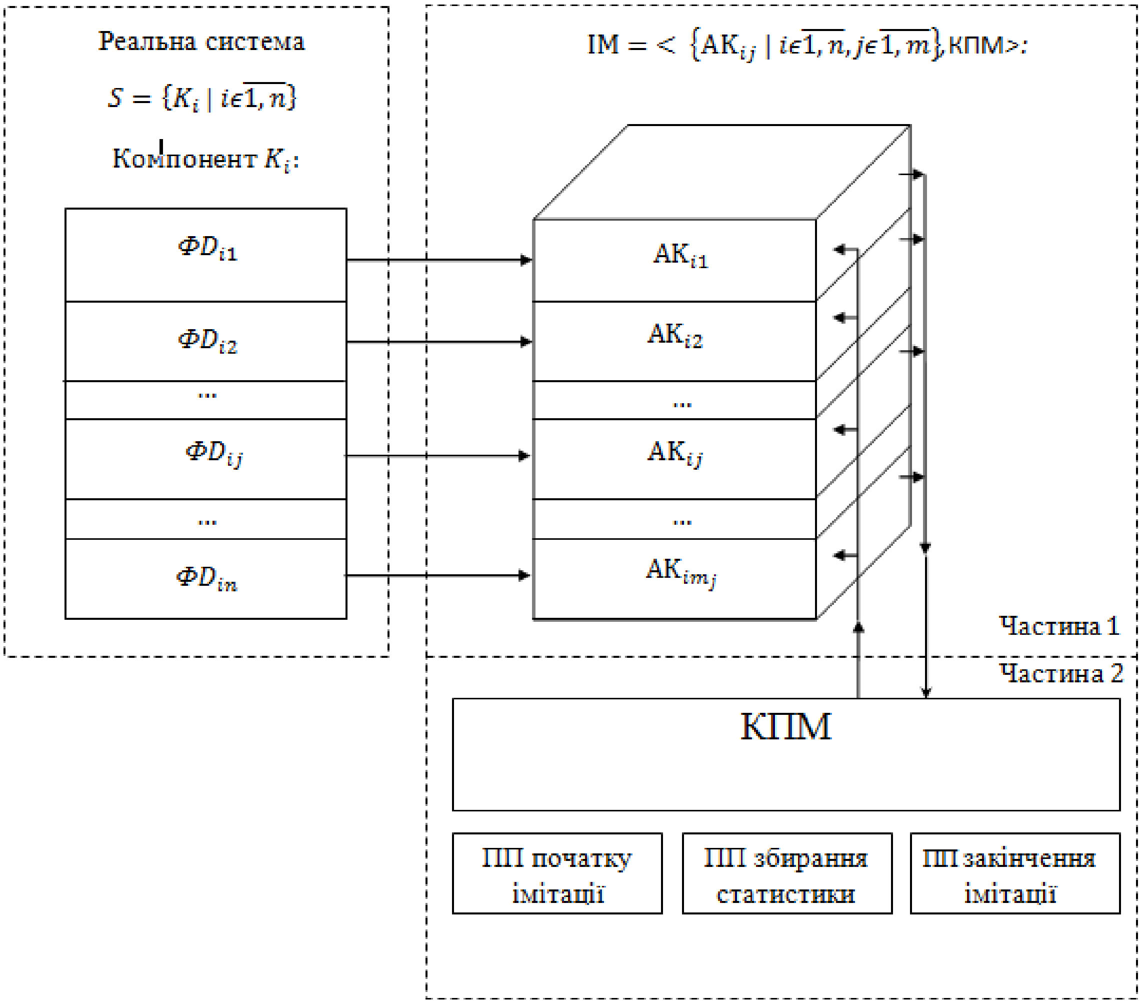


Рисунок 4.8 – Структурна схема імітаційної моделі

На рис. 4.8 наочно показано, що кожна функціональна дія $ФD_{ij}$ компоненти K_i в імітаційній моделі описується відповідною активністю $АК_{ij}$, яка подана алгоритмом $АЛ_{ij}$ і оператором Mt_{ij}, що змінює локальну координату t_i на значення тривалості виконання дії τ_{ij}. Зі схеми випливає, що команди на запуск виконання алгоритмів виходять від КПМ. Після виконання операторів Mt_{ij} інформація про нові значення термінів настання подій C_{ij} надходить у КПМ і використовується для модифікації модельного часу t_0 у формуванні нових команд на запуск алгоритмів АЛ_{ij}, у перевірці умов закінчення імітації.

У процесі розроблення імітаційної моделі виділяються наступні взаємопов'язані етапи:

- визначення цілей моделювання;
- розробка концептуальної моделі;
- розробка програмної реалізації імітаційної моделі;
- перевірка здійсненності вимог до моделі;
- планування і проведення експериментів;
- аналіз результатів моделювання.

Цілі моделювання визначаються цілями дослідження, які найчастіше полягають у створенні нової системи або в модернізації функціонуючої системи з урахуванням ефективності.

Показником ефективності є перевищення корисності пропонованого варіанта системи над вартістю створення й експлуатації цього варіанта. Проте визначити ефект у повному обсязі, як правило, не вдається, оскільки він проявляється не тільки безпосередньо, як результат функціонування пропонованого варіанта системи, а й опосередковано, як результат функціонування інших систем, з якими пов'язана досліджувана система.

Більш простим є поняття техніко-економічної ефективності, яке включає витрати і вимірювані характеристики системи чи інакше – часткові показники якості системи: продуктивність, надійність, маса, габарити і т.д. Ці часткові показники є компонентами вектора в моделі (1.1), а сам показник техніко-економічної ефективності є функцією від Y

$$E = E(Y). \tag{4.2}$$

Якщо оцінка ефективності (4.2) зв'язується з оптимізацією за одним показником якості і з обмеженням інших показників, наприклад,

$$E = y_j \to \min\,(\max);$$

$$y_i \begin{pmatrix} > \\ < \\ = \end{pmatrix} y_i^0, \ \ (i \in \overline{1,n}, \ \ i \neq j),$$

де y_j – компоненти вектора Y, а y_i^0 – їхні деякі кількісні значення, то ця оцінка ефективності є однокритеріальною.

Багатокритеріальною оцінкою є узагальнений або інтегральний критерій, який пов'язує функціональною залежністю показник ефективності E з перетвореними компонентами вектора $Y - y'_j$ $(i \in \overline{1,n})$. Перетворення компонент дозволяє привести їх до безрозмірного вигляду і зіставляти відхилення компонент від своїх оптимальних значень.

Найбільш широко використовується узагальнений критерій у вигляді лінійної функції від перетворених компонент:

$$E = \sum_{i \in \overline{1,n}} \alpha_i y'_i\,, \tag{4.3}$$

де $\forall i \in \overline{1,n}$ $\; 0 \le \alpha_i \le 1,\; \sum\limits_{i \in \overline{1,n}} \alpha_i = 1$.

Коефіцієнти $\alpha_i (i \in \overline{1,n})$ відображають важливість або «вагу» відповідних компонент $y'_i (i \in \overline{1,n})$.

Таким чином, цілі моделювання полягають, по-перше, у визначенні працездатного варіанта системи, у якому вихідні характеристики (параметри) задовольняють заданим обмеженням

$$Y \begin{pmatrix} < \\ > \\ = \end{pmatrix} Y_0 \tag{4.4}$$

і, по-друге, у визначенні оптимального варіанта системи з точки зору заданого критерію ефективності, що зводиться до постановки та розв’язання відповідної задачі однокритеріальної або багатокритеріальної оптимізації.

Для досягнення будь-якої з зазначених цілей необхідно визначити внутрішні параметри, зміна значень яких забезпечує необхідні значення вихідних параметрів.

Розробку концептуальної (змістовної) моделі пов’язують із створенням абстрактної моделі, за допомогою якої визначається склад і структура досліджуваної системи S', властивості елементів і причинно-наслідкові зв’язки, властиві S' і суттєві для досягнення мети моделювання. Таким чином, на цьому етапі система реального світу, яку необхідно вивчити, представляється у вигляді концептуальної моделі, тобто математичних і логічних відносин, що зв’язують компоненти у структуру системи. Рекомендується починати з простої моделі даної системи і у міру вивчення системи розширювати модель до необхідного рівня складності.

Етап розроблення концептуальної моделі реалізується виконанням таких дій, як орієнтація, стратифікація, локалізація, структуризація і керування, виділення процесів.

Під орієнтацією розуміється уявний розгляд досліджуваної системи S' у рамках мегасистеми, частиною якої є S'. Виділяючи систему S' з мегасистеми, дослідник повинен обґрунтувати з точки зору цілей моделювання, які властивості і зв'язки системи зберігаються в моделі, а які не враховуються як несуттєві.

Стратифікація – це процес виділення рівнів деталізації системи S', які називають стратами.

Як показано на рис. 4.9, на найвищому n-му рівні розташовується система, яку можна вважати підсистемою n-го рівня. Нижче розташовуються підсистеми $(n-1)$ рівня, які безпосередньо утворюють систему. Ще нижче розташовуються підсистеми $(n-2)$ рівня, які утворюють підсистеми вищих рівнів. І так далі, нарешті, на найнижчому рівні розташовуються підсистеми нульового рівня або інакше – елементи системи. Зазвичай деталізацію проводять до рівня, для якого відомі залежності вихідних характеристик елементів від параметрів.

Модель складної системи у вигляді зазначеного на рис. 4.9 стратифікованого подання є сімейством моделей, кожна з яких відображає поведінку досліджуваної системи на різних рівнях деталізації. При цьому на кожному рівні можуть використовуватися відповідні принципи, змінні і залежності.

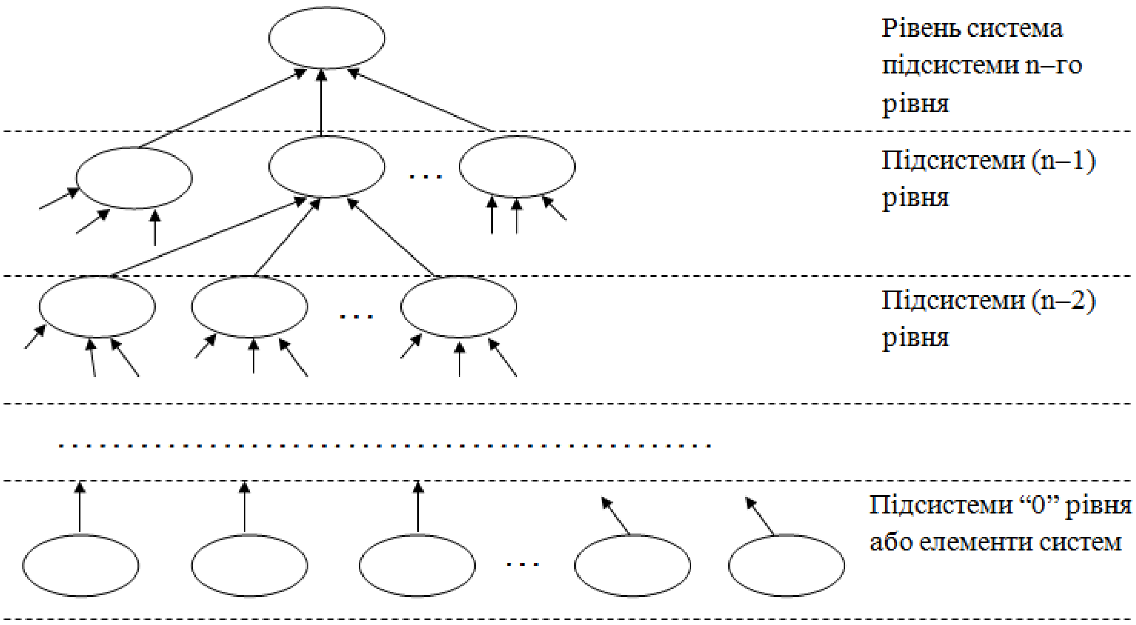


Рисунок 4.9 – Стратифіковане уявлення складної системи

Локалізація реалізується поданням зовнішнього середовища, сукупністю генераторів зовнішніх впливів. Ці генератори, як правило, є елементами моделі системи.

Структуризацією передбачається визначення всіх елементів моделі і зв'язків між ними, які можуть бути речовими та інформаційними. Речові елементи відображають переміщення продукту перетворення між елементами, а інформаційні призначаються для передачі керуючих впливів і відомостей про стан елементів.

У деяких простих системах можуть бути відсутні інформаційні зв'язки. Управління функціонуванням такими системами реалізується згідно з принципом структурного управління, тобто визначається структурою системи.

Однак найчастіше в системах є інформаційні зв'язки і вирішальні елементи, в яких виробляються керуючі впливи. В управліннях є вказівки щодо того, якому елементу і звідки взяти вихідний об'єкт, що з ним зробити і, нарешті, куди передати перетворений об'єкт. Такі системи функціонують згідно з програмним або алгоритмічним принципом управління. У концептуальній моделі таких систем мають бути визначені всі вирішальні правила або алгоритми управління.

Виділення процесів виконується для моделювання динамічних систем. Функціонування цих систем відображається описом виконання технологічних процесів перетворення речовини, енергії або інформації. Технологічний процес при цьому подається одним з видів алгоритму, що визначає, які операції, в якій послідовності і за допомогою яких ресурсів необхідно виконати для досягнення системою заданого цільового стану. Сукупність алгоритмів, що відображають реалізацію всіх виділених у системі технологічних процесів, є по суті імітаційної моделлю досліджуваної системи.

Розробка програмної реалізації імітаційної моделі реалізується на основі обраних програмних засобів, які поділяються на процедурно-орієнтовані, проблемно-орієнтовані мови й автоматизовані системи моделювання.

Після вибору мови розробляється програмна модель, що передбачає виконання таких відомих дій, як уточнення і деталізація алгоритму, визначення форм вхідних даних і результатів, написання і налагодження програми.

У разі використання автоматизованої системи моделювання програмна модель створюється автоматично на основі заданої математичної схеми, вихідних параметрів системи, що задаються дослідником, зовнішнього середовища і особливостей функціонування системи.

Перевірку здійсненності вимог до моделі на основі співвідношення (1.2) для багатьох складних систем реалізувати важко з кількох причин. По-перше, виникають труднощі в отриманні вектора вихідних параметрів системи Y_S, коли система є проектованою або модернізується, тобто, коли її фактично немає. По-друге, (1.2) це векторна оцінка, і її виконання досить складно домогтися для всіх компонент вектора без використання схем компромісу. По-третє, і вектор вихідних параметрів моделі Y_M, і системи Y_S можуть бути випадковими, нестаціонарними функціями, і тому для отримання їх характеристик необхідно обробити досить великий обсяг статистик моделювання. По-четверте, необхідно обґрунтувати значення граничних відхилень відповідних компонент векторів Y_M і Y_S.

На практиці перевірку здійсненності вимог до моделей реалізують за допомогою кількох перевірок, результати яких аналізуються експертами. Перевіркам піддаються моделі всіх структурних елементів і зв'язків між ними, модель зовнішнього середовища для досліджуваної системи і т.д.

Найчастіше виконуються наступні види перевірок:

- перевірка моделей елементів досліджуваної системи, результатом якої може бути подальша деталізація елементів;
- перевірка моделей зовнішніх впливів, результатом якої може бути зміна припущень, гіпотез щодо характеру впливів;
- перевірка концептуальної моделі, результати якої можуть змінити постановку задачі;
- перевірка прийнятої математичної схеми на відповідність характеру процесів у досліджуваній системі;
- перевірка способів вимірювання і обчислення параметрів, що дозволяє виявити значення відповідних похибок;
- перевірка програмної реалізації моделі.

Якщо за результатами перевірок виявляється неприпустима неузгодженість моделі та досліджуваної системи, то виникає необхідність у коригуванні чи калібруванні моделі. Виділяються три основних види змін: глобальні, локальні і параметричні.

Глобальні зміни проводяться при виявленні у концептуальній моделі методичних помилок, які усуваються шляхом побудови нової моделі.

Локальні зміни пов'язуються з розробкою нових моделей окремих елементів, зв'язків або впливів.

До параметричних змін відносять зміни значень деяких, так званих калібрувальних параметрів у заздалегідь встановлених межах.

Перевірка здійсненності вимог до моделі реалізується на основі процесів верифікації та валідації моделі. *Верифікація* стосується програмної реалізації імітаційної моделі. Вона включає в себе перевірку правильності функціонування моделі. Рекомендується, щоб процес верифікації був неперервним. При цьому доцільно використовувати контролери та відладчики, що надаються засобами автоматизованої розробки імітаційних моделей.

Валідація дозволяє упевнитися, що розроблена концептуальна модель дійсно є поданням реальної системи. Процес валідації націлений на перевірку того, чи можна замінити реальну систему розробленою моделлю в цілях дослідження. Якщо реальна система існує, то найкращим способом виконати валідацію буде порівняння виходів моделі з виходами реальної системи. Однак при розробленні нової системи бази для порівняння не існує, тому дослідники користуються іншими способами валідації.

Планування та проведення експериментів реалізується з метою отримання статистик моделювання, обробка яких дозволяє визначити шукані параметри.

План експериментів являє собою певний порядок наборів поєднань значень варійованих параметрів (факторів). Число наборів визначає число експериментів.

Основне завдання планування експериментів, що проводяться з моделлю на комп’ютері, полягає в отриманні всієї необхідної інформації про досліджувану систему при мінімальних або обмежених витратах ресурсів комп’ютера на реалізацію процесу отримання статистик моделювання.

Аналіз результатів моделювання реалізується після обробки статистик моделювання методами кореляційного, дисперсійного та регресійного аналізу, що дозволяє отримати залежність (1.1), де F може бути функцією або алгоритмом.

До аналізу результатів моделювання належить завдання аналізу чутливості моделі до варіацій її параметрів. Зокрема, перевіряється стійкість вихідних параметрів до можливих змін параметрів системи і зовнішнього середовища.

На основі аналізу результатів моделювання уточнюються параметри моделі, що призводить до корекції концептуальної моделі, шукається можливість створення аналітичної моделі досліджуваної системи або визначаються вагові коефіцієнти у критерії ефективності (4.3).

Обов’язковою умовою успішного імітаційного дослідження є документування і складання звітів. Якщо імітаційна модель буде повторно використовуватися тим же або іншим аналітиком, можливо знадобиться повернутися до розуміння того, як вона побудована і функціонує. Документовані звіти підвищать впевненість при прийнятті рішень. Також документація вкрай корисна в разі, якщо модель потребує змін. Звіти про результати імітаційного дослідження необхідні для прийняття рішень, що стосуються модельованої системи.

У кінцевому підсумку результати моделювання використовуються для прийняття рішення про працездатність системи у випадку, коли визначаються параметри, що задовольняють системі обмеження (4.4), або про вибір оптимального варіанта з множини допустимих з урахуванням критерію, який виражає ефективність (4.2).

4.4. Планування експериментів у методі імітаційного моделювання

З визначення імітаційного моделювання випливає важлива роль, яка відводиться організації та проведенню експериментів з імітатором для отримання статистик моделювання, що характеризують поведінку компонент складної системи в різних режимах функціонування.

Основне завдання планування експериментів з імітатором полягає в отриманні всієї необхідної інформації про об’єкт моделювання при мінімальних або обмежених витратах ресурсів комп’ютера на реалізацію процесу моделювання.

Це завдання розв’язується підпрограмою збbhfyyz статистики моделювання КПМ на основі теорії планування експериментів, ключовими поняттями якої є поняття фактора і реакції [25, 26].

Якщо метою експерименту є вивчення впливу змінної x на змінну y, то x називається фактором, а y – реакцією. Кожен фактор у процесі експерименту приймає різні значення, які називаються рівнями.

Кожному фіксованому набору рівнів факторів відповідає певна точка в багатовимірному просторі, який називається факторним простором.

Положення точки у факторному просторі визначається набором рівнів факторів $(x_1, x_2, ..., x_k)$ і значенням реакції y, яка зв’язується з набором рівнів за допомогою функції реакції $\Psi -\ y = \Psi(x_1, x_2, ..., x_k)$.

Геометричне місце з m точок $y_l = \Psi(x_{1l}, x_{2l}, ..., x_{kl})$ $(l \in \overline{1, m})$ утворює геометричний образ поверхні реакції.

Вид (структура) функції Ψ та її параметри часто є математичною моделлю досліджуваної системи. Для їх визначення проводяться експерименти, які повинні бути так організовані, щоб при мінімальних витратах ресурсів (наприклад, мінімальному числі експериментів) отримати правила варіювання рівнями факторів.

У роботі [6] наводиться досить повна класифікація факторів і вимог до них, які необхідно враховувати при складанні правил варіювання рівнями факторів, що і є по суті планом експериментів.

Фактори можуть бути керованими і некерованими, що спостерігаються і неспостережуваними, досліджуваними і такими, що не вивчаються, кількісними та якісними, фіксованими і випадковими.

Фактор є керованим, якщо його рівні цілеспрямовано вибираються в процесі експерименту. В іншому випадку фактор є некерованим.

Фактор є спостережуваним, якщо його рівні спостерігаються і реєструються. В іншому випадку фактор є неспостережуваним.

Спостережувані, але некеровані чинники називаються супутніми. Число таких є значним, і тому серед них враховуються тільки ті, які суттєво впливають на реакцію.

Фактор є досліджуваним, якщо він включається в модель для вивчення властивостей об'єкта. В іншому випадку фактор є такими, що не вивчається.

Фактор є кількісним, якщо його рівні представляються числовими величинами. Наприклад, у неперервно-стохастичних моделях, створюваних на основі теорії систем масового обслуговування, кількісними чинниками є інтенсивність вхідних потоків заявок на обслуговування, час обслуговування заявок каналом, а якісними факторами є правила постановки заявок в чергу перед початком обслуговування, порядок призначення каналу на обслуговування і т.д.

Фактор є фіксованим, якщо розглядаються всі його рівні. Якщо ж береться деяка випадкова вибірка рівнів фактора із сукупності можливих рівнів, то фактор є випадковим.

Імітатор вважається визначеним, якщо визначена вся множина істотних факторів і вона не змінюється у процесі даного експерименту.

Планом експерименту зазвичай передбачається одночасна зміна кількох чинників. Основними вимогами до сукупності одночасно змінюваних факторів є їх сумісність і незалежність.

Сумісність означає, що всі комбінації (поєднання) факторів здійсненні, а незалежність відповідає можливості встановити фактор на будь-якому рівні з множини допустимих незалежно від рівня інших факторів із сукупності одночасно змінюваних.

У теорії планування експериментів функція $\Psi(x_1, x_2, ..., x_k)$ часто подається у вигляді полінома ступеня d від k факторів:

$$y = \hat{b}_0 + \sum_{1 \le i \le k} \hat{b}_i x_i + \sum_{1 \le i, j \le k} \hat{b}_{ij} x_i x_j + \ldots + \sum_{1 \le i_1, i_2, ..., i_k \le k} b_{i_1, i_2, ..., i_k} x_1^{i_1} x_2^{i_2} .. x_k^{i_k}, \tag{4.5}$$

де $\sum_{1 \le j \le k} i_j = d$.

Поліном (4.5) має $C_{d+k}^{d} = \dfrac{(d+k)!}{d!k!}$ коефіцієнтів, які необхідно визначити на основі результатів експериментів.

Якщо $x_0 = 1$, $x_{k+1} = x_1^2$, $x_{k+2} = x_2^2$, $x_{2k+1} = x_k^2$, $x_{2k+2} = x_1 x_2$, $x_{2k+3} = x_1 x_3$, …, $x_{k^1} = x_{k-(d-1)} x_{k-(d-2)} .. x_{k-1} x_k$, де у виразі для x_k d співмножників, то вираз (4.5) подається однорідним лінійним рівнянням

$$y = \sum_{0 \le \alpha \le k'} \hat{b}'_\alpha x_\alpha , \tag{4.6}$$

де $\hat{b}'_0 = \hat{b}_0$, $\hat{b}'_1 = \hat{b}_1$, …, $\hat{b}'_k = \hat{b}_k$, $\hat{b}'_{k+1} = \hat{b}_{11}$, …, $\hat{b}'_{2k+1} = \hat{b}_{kk}$, $\hat{b}'_{2k+2} = \hat{b}_{12}$, …, $\hat{b}'_{k'} = \hat{b}_{(k-(d-1))\dots(k-1)k}$.

Процес перетворення виразу (4.5) у вираз (4.6) покажемо на прикладі двофакторної моделі.

У цьому випадку вираз (4.5) має вигляд

$$y = \hat{b}_0 + \hat{b}_1 x_1 + \hat{b}_2 x_2 + \hat{b}_{11} x_1^2 + \hat{b}_{22} x_2^2 + \hat{b}_{12} x_1 x_2 . \tag{4.7}$$

Якщо припустити, що $x_0 = 1$ і до наявних позначень двох факторів x_1, x_2 додати нові чинники, які позначають нелінійні елементи в розглянутій двофакторній моделі, тобто

$$x_3 = x_1^2 ,\ x_4 = x_2^2 ,\ x_5 = x_1 x_2 ,$$

то отримуємо наступний вираз вигляду (4.6) для моделі (4.7)

$$y = \sum_{0 \le \alpha \le 5'} \hat{b}'_\alpha x_\alpha , \tag{4.8}$$

де $\hat{b}'_0 = \hat{b}_0$, $\hat{b}'_1 = \hat{b}_1$, $\hat{b}'_2 = \hat{b}_2$, $\hat{b}'_3 = \hat{b}_3$, $\hat{b}'_4 = \hat{b}_{22}$, $\hat{b}'_5 = \hat{b}_{12}$.

Перевага виразу (4.6) перед (4.5) полягає в його лінійності, що спрощує отримання коефіцієнтів $\hat{b}'_\alpha$ у порівнянні з отриманням коефіцієнтів $\hat{b}'_i$. Однак при цьому необхідно мати на увазі, що підвищується розмірність задачі, тому що число визначених коефіцієнтів зростає ($k' > k$).

Поліном (4.5) містить C_{d+k}^d коефіцієнтів, які необхідно визначити на основі результатів експериментів. Тому план експериментів *D* повинен містити число точок *N*, в яких проводиться експеримент, що не менше числа визначених коефіцієнтів, тобто $N \ge C_{d+k}^d$.

План *D* представляється матрицею з рядками, що відповідають точкам факторного простору, в яких проводяться експерименти. Ця матриця має наступний вигляд:

$$D = \begin{Vmatrix} x_{11} & x_{21} & \dots & x_{k1} \\ x_{12} & x_{22} & \dots & x_{k2} \\ \dots & \dots & \dots & \dots \\ x_{1N} & x_{2N} & \dots & x_{kN} \end{Vmatrix}, \tag{4.9}$$

де x_{in} ($i \in \overline{1,k}$, $n \in \overline{1,N}$) – рівень *i*-го фактора в експерименті *n*.

Реалізація експериментів у *N* точках факторного простору, координати яких записуються в *N* рядках матриці *D*, дозволяє отримати вектор спостережень (результатів експериментів): $y = \begin{Vmatrix} y_1 & y_2 & \dots & y_N \end{Vmatrix}^T$, де

y_n $(n \in \overline{1,N})$ – реакція, що відповідає експерименту n у точці плану $\left\| x_{1n} \quad x_{2n} \quad \dots \quad x_{kn} \right\|$.

Якщо модель, що зв'язує реакцію і фактори, визначається виразом (4.6), то план експериментів X повинен містити N точок, число яких також має бути не менше числа певних коефіцієнтів, тобто $N \geq k' + 1$.

План X, як і план D, представляється матрицею з рядками, що відповідають точкам факторного простору, в яких проводяться експерименти. Матриця, яка визначає план X, має наступний вигляд:

$$X = \begin{Vmatrix} x_{01} & x_{11} & x_{21} & \dots & x_{k1} & x_{k+1\,1} & \dots & x_{k'1} \\ x_{02} & x_{12} & x_{22} & \dots & x_{k2} & x_{k+1\,2} & \dots & x_{k'2} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{0N} & x_{1N} & x_{2N} & \dots & x_{kN} & x_{k+1\,N} & \dots & x_{k'N} \end{Vmatrix}, \tag{4.10}$$

де $x_{on} = 1$ $(n \in \overline{1,N})$, x_{kn} $(k \in \overline{1,k'}, n \in \overline{1,N})$ – рівень фактора k в експерименті n.

Експеримент n, проведений в точці факторного простору, координати якої записуються в n-му рядку матриці $X - \left\| x_{0n} \quad x_{1n} \quad x_{2n} \quad \dots \quad x_{kn} \quad \dots \quad x_{k'n} \right\|$, дозволяє визначити відповідне значення реакції

$$y_n = \hat{b}_0' x_{0n} + \hat{b}_1' x_{1n} + \hat{b}_2' x_{2n} + \ldots + \hat{b}_k' x_{k'n} + e_n,$$

де e_n – помилка експерименту, яка передбачається незалежною, нормально розподіленою, випадковою величиною з математичним очікуванням $M[e_n] = 0$ і постійною дисперсією – $D[e_n] = \text{const}$.

Вибір рівнів факторів у планах експериментів D і X реалізується з урахуванням наступних рекомендацій.

Спочатку вибираються границі $x_{i\min}$ і $x_{i\max}$ області визначення факторів. Вибір реалізується на основі фізичних міркувань щодо властивостей досліджуваного об'єкта. Наприклад, якщо фактором є напруга струму побутової мережі, то $x_{i\min} = 200\,\text{В}$, а $x_{i\max} = 250\,\text{В}$. Обрані межі визначають деяку локальну область G факторного простору.

Для кожного фактора далі встановлюється основний або нульовий рівень x_{i0} та інтервал варіювання Δx_i. Наприклад, для лінійної двофакторної моделі

$$y = b_0 + b_1 x_1 + b_2 x_2$$

область G з нульовими рівнями факторів зображена на рис. 4.10.

На рис. 4.10 показано, що план експериментів будується шляхом варіювання кожного фактора x_i на декількох рівнях щодо вихідної точки x_{i0}, яка визначає центр експериментів.

План експериментів, в якому реалізуються всі можливі сполучення рівнів факторів, називається повним факторним експериментом (ПФЕ).

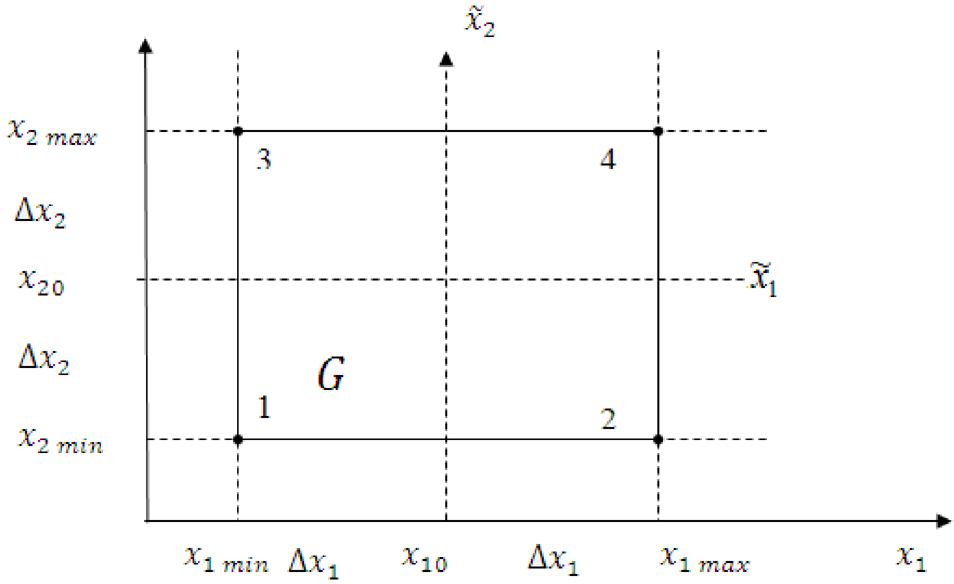


Рисунок 4.10 – Область G з різними поєднаннями рівнів у 1-4-му експериментах

У тому випадку, коли структура моделі включає тільки лінійні члени полінома і добутки факторів, для оцінення коефіцієнтів моделі використовується план експериментів з варіюванням усіх k факторів на двох рівнях. Такі плани називаються планами 2^k, де $N = 2^k$ – число всіх можливих експериментів для оцінення коефіцієнтів моделі зазначеної структури.

Наприклад, нехай $k = 3$. Тоді трьохфакторна модель, структура якої включає тільки лінійні члени та добутки факторів, має наступний вигляд:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{12}x_1x_2 + b_{13}x_1x_3 + b_{23}x_2x_3 + b_{123}x_1x_2x_3.$$

Оцінки потрібні для восьми коефіцієнтів зазначеної моделі – b_0, b_1, b_2, b_3, b_{12}, b_{13}, b_{23}, b_{123}. Для їх оцінення потрібно провести $N = 2^k = 2^3 = 8$ експериментів. Легко бачити, що число експериментів строго відповідає числу визначених коефіцієнтів. Зауважимо ще, що це число експериментів є мінімальним для визначення коефіцієнтів у розглянутій моделі.

У реальних задачах фактори мають різні одиниці виміру і діапазон (інтервал) зміни. Для утворення поєднань таких факторів необхідно попередньо провести операцію масштабування, для чого часто використовується перетворення

$$\hat{x}_i = \frac{x_i - x_{i0}}{\Delta x_i} \; (i \in \overline{1,k}), \tag{4.11}$$

де $\hat{x}_i$ – масштабований рівень i-го фактора; x_i – натуральний рівень i-го фактора; а Δx_i визначається виразом

$$\Delta x_i = \frac{x_{i\max} - x_{i\min}}{2}.$$

У планах 2^k, як відомо, кожен фактор варіюється на двох рівнях. Зазвичай цими рівнями є $x_{i\min}$ та $x_{i\max}$ $(i \in \overline{1,k})$. На основі перетворення (4.11) легко отримати, що рівню $x_{i\min}$ відповідає масштабований рівень $\hat{x}_i = -1$, а рівню $x_{i\max}$ $-\hat{x}_i = +1$.

Масштабована область G плану 2^2 зображена на рис. 4.11.

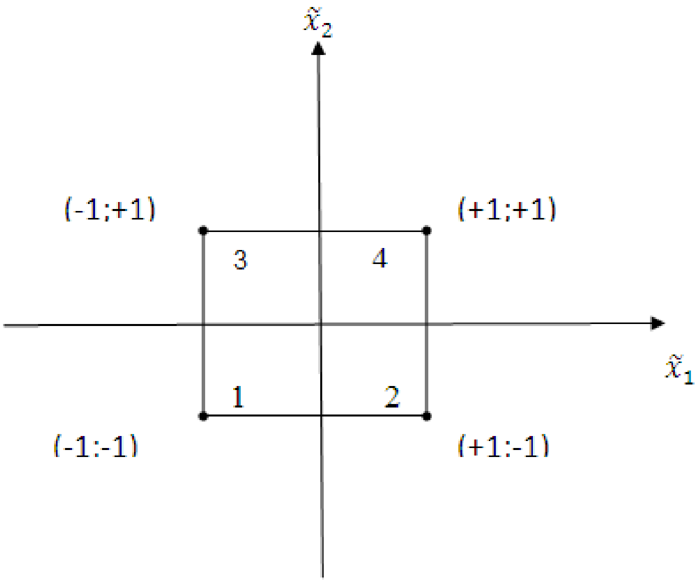


Рисунок 4.11 – Область G плану 2^2

Показаний на рис. 4.11 план 2^2 у вигляді вершин області G, як уже зазначалося, подається матрицею планування D, яка має наступний вигляд:

$$D = \begin{vmatrix} -1 & -1 \\ +1 & -1 \\ -1 & +1 \\ +1 & +1 \end{vmatrix}.$$

У матриці D номер рядка відповідає номеру експерименту плану.

Число експериментів плану 2^2 цілком достатньо для оцінки чотирьох коефіцієнтів моделі виду

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_{12} x_1 x_2.$$

Якщо для опису поверхні реакції обмежитися лінійною моделлю

$$y = b_0 + b_1 x_1 + b_2 x_2,$$

в якій необхідно оцінити тільки три коефіцієнти b_0, b_1, b_2, то, очевидно, план 2^2 є надлишковим, оскільки число експериментів цього плану більше числа оцінюваних коефіцієнтів на одиницю.

Легко показати, що вказана надмірність зростає із збільшенням числа факторів *k*.

Дійсно, при $k = 3$ число експериментів плану 2^k дорівнює $N = 2^3 = 8$, а число оцінюваних коефіцієнтів лінійної трьохфакторної моделі

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3$$

дорівнює 4. Тобто надмірність при $k = 3$ становить чотири. Зауважимо, що при $k = 2$, ця надмірність дорівнювала одиниці.

Оскільки проведення експериментів вимагає витрат ресурсів, то надмірність призводить до неефективного їх використання.

З метою мінімізації надлишковості поряд з ПФЕ виду 2^k, коли кожен фактор варіюється на двох рівнях, розглядають дробові факторні експерименти, число яких відповідає числу оцінюваних коефіцієнтів моделі, що описує поверхню реакції.

У теорії планування експериментів поряд з симетричними дворівневими планами 2^k використовуються багаторівневі плани, в яких фактори варіюються на 3, 4 і т.д., *m* рівнях. Ці плани позначаються відповідно 3^k, 4^k, …, m^k.

Крім цього, в теорії планування експериментів використовуються також багаторівневі несиметричні плани, в яких фактори варіюються на різних рівнях. Число експериментів у таких планах визначається як

$$N = q_1 q_2 .. q_k , \tag{4.12}$$

де q_i ($i \in \overline{1,k}$) – число рівнів *i*-го фактора.

Зменшення числа експериментів *N*, а значить, і витрат ресурсів на їх проведення, можливо шляхом зменшення числа співмножників у виразі (4.12), що рівносильно зменшенню числа факторів *k*, а також шляхом зменшення значень співмножників q_i ($i \in \overline{1,k}$), що рівносильно зменшенню числа рівнів *i*-го фактора.

Число факторів визначається метою експериментів. Як правило, розглядаються дві основні мети. Перша з них полягає в побудові залежності реакції від факторів для виявлення властивостей системи, що пов’язується з розв’язанням певної задачі аналізу. Друга мета полягає у знаходженні деякої комбінації рівнів факторів, яка надає екстремальне значення реакції, що пов’язується з розв’язанням задачі синтезу.

Для досягнення першої мети, як правило, потрібна реалізація ПФЕ. Однак при цьому необхідно враховувати істотні фактори. З практики відомо, що 20 % факторів визначають 80 % властивостей системи, а інші 80

% чинників визначають решту 20 % властивостей системи. Ці 20 % факторів і є суттєвими. Вони визначаються на основі експериментів.

Для досягнення другої мети використання ПФЕ не є раціональним, оскільки це рівносильно повному перебору варіантів, що призводить до великих витрат ресурсів. Для досягнення другої мети більш раціональним є отримання апріорної інформації про характер поверхні реакції і на основі цієї інформації використання одного з вибіркових методів пошуку оптимуму поверхні реакції. При цьому можна скористатися такими добре відомими методами, як методи випадкової вибірки, рівномірної сітки, найшвидшого спуску та ін. [27].

4.5. Точність, достовірність результатів моделювання та правила автоматичної зупинки експерименту

Проблема забезпечення точності та достовірності результатів моделювання виникає, коли в імітаційній моделі є елементи статистичного моделювання, що призводить до стохастичності результатів експериментів з моделлю.

Рішення проблеми пов'язується з отриманням оцінок точності і достовірності при заданому числі експериментів, яке обумовлюється обмеженими ресурсами, або, навпаки, з отриманням оцінки числа експериментів при заданих значеннях точності та достовірності.

Нехай E – деякий показник досліджуваної системи S, значення якого є результатом експерименту. Позначимо $\hat{E}$ оцінку розглянутого показника.

Стохастичність результатів експериментів і обмеженість числа експериментів N призводять до того, що в загальному випадку $E \neq \hat{E}$.

Величина $\varepsilon = \left| E - \hat{E} \right|$ називається точністю оцінки, а ймовірність $Q = P\left\{ \left| E - \hat{E} \right| < \varepsilon \right\}$ – її достовірністю.

Величину $\varepsilon_0 = \dfrac{\varepsilon}{E}$ називають відносною точністю, а ймовірність

$$Q_0 = P\left\{ \frac{\left| E - \hat{E} \right|}{E} < \varepsilon \right\}$$

– відносною достовірністю оцінки.

Для того щоб за заданими ε, Q визначити число експериментів N або, навпаки, за заданим значенням N знайти точність ε і достовірність Q, необхідно знати закон розподілу, принаймні, величини точності ε. Однак апріорі він невідомий. Дана складність долається висуванням припущення про характер закону розподілу випадкової величини оцінки $\hat{E}$.

Наприклад, нехай метою експериментів з імітаційної моделлю є отримання оцінки $\hat{p}$ ймовірності появи деякої події A – $p = p(A)$. В даному випадку $E = p$, $\hat{E} = \hat{p}$.

Оцінка $\hat{p}$ може розглядатися як $\hat{p} = \dfrac{m}{N}$, де m – число появ події A в N експериментах.

Тоді достовірність Q визначається виразами:

$$Q = P\left\{\left|P - \frac{m}{N}\right| < \varepsilon\right\};$$

$$Q = P\left\{p - \varepsilon < \frac{m}{N} < p + \varepsilon\right\}.$$

Оцінку $\hat{p}$ можна подати як

$$\hat{p} = \frac{1}{N}\sum_{i \in \overline{1,N}} x_i \,,$$

де x_i – результат реалізації деякої випадкової величини ξ, яка приймає значення 1, коли подія A відбувається, і значення 0, коли подія A не відбувається. При цьому $x_i = 1$ з імовірністю p, а $x_i = 0$ з ймовірністю $(1 - p)$
.

Визначимо математичне очікування $M[\xi]$ і дисперсію $D[\xi]$ випадкової величини ξ:

$$M[\xi] = x_i p + x_i(1 - p) = 1 \cdot p + 0 \cdot (1 - p) = p;$$

$$D[\xi] = \left(x_i - M[\xi]\right)^2 p + \left(x_i - M[\xi]\right)^2 (1 - p) = (1 - p)^2 p + (0 - p)^2 (1 - p) =$$
$$= p(1 - p).$$

Отримані значення $M[\xi] = p$, $D[\xi] = p(1 - p)$ використовуємо для визначення математичного очікування і дисперсії оцінки $\hat{p}$ – $M[\hat{p}], D[\hat{p}]$:

$$M[\hat{p}] = M\left[\frac{m}{N}\right] = M\left[\frac{1}{N}\sum_{i \in \overline{1,N}} x_i\right] = \frac{1}{N} M\left[\sum_{i \in \overline{1,N}} x_i\right] = \frac{1}{N}\sum_{i \in \overline{1,N}} M[x_i] = \frac{1}{N} N \cdot M[\xi] =$$
$$= M[\xi] = p;$$

$$D[\hat{p}] = D\left[\frac{m}{N}\right] = \frac{1}{N^2} D\left[\sum_{i \in \overline{1,N}} x_i\right] = \frac{1}{N^2}\sum_{i \in \overline{1,N}} D[x_i] = \frac{1}{N^2} N \cdot D[\xi] = \frac{1}{N} p(1 - p).$$

Те, що $M[\hat{p}] = p$, означає незсуненість оцінки $\hat{p}$.

З центральної граничної теореми теорії ймовірностей випливає, що $\hat{p} = \dfrac{m}{N}$ за досить великих значень N можна розглядати як випадкову величину, яка описується нормальним законом розподілу ймовірностей з математичним очікуванням $M[\hat{p}] = p$ та дисперсією $D[\hat{p}] = \frac{1}{N} p(1-p)$.

Тому для визначення достовірності оцінки скористаємося співвідношенням:

$$Q = P\left\{p - \varepsilon < \frac{m}{N} < p + \varepsilon\right\} = \Phi_0\left(\frac{p+\varepsilon-p}{\sqrt{p(1-p)}}\sqrt{N}\right) - \Phi_0\left(\frac{p-\varepsilon-p}{\sqrt{p(1-p)}}\sqrt{N}\right),$$

де Φ_0 – стандартна функція, що зветься інтегралом Лапласа.

Виходячи з властивості симетрії кривої нормального закону розподілу, що означає

$$\Phi_0(-z) = 1 - \Phi_0(z),$$

для розглянутого випадку отримуємо:

$$\Phi_0\left(\frac{p-\varepsilon-p}{\sqrt{p(1-p)}}\sqrt{N}\right) = 1 - \Phi_0\left(\frac{p+\varepsilon-p}{\sqrt{p(1-p)}}\sqrt{N}\right).$$

З урахуванням цього співвідношення вираз для достовірності оцінки набуває вигляду:

$$Q = \Phi_0\left(\frac{p+\varepsilon-p}{\sqrt{p(1-p)}}\sqrt{N}\right) - 1 + \Phi_0\left(\frac{p+\varepsilon-p}{\sqrt{p(1-p)}}\sqrt{N}\right).$$

Елементарні перетворення дозволяють отримати співвідношення

$$\Phi_0\left(\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}}\right) = \frac{1+Q}{2}.$$

З чого випливає, що

$$\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}} = t_\varphi, \tag{4.13}$$

де t_φ – квантиль нормального закону розподілу ймовірностей порядку

$$\varphi = \frac{1+Q}{2},$$

значення якого наводяться у спеціальних таблицях.

З виразу (4.13) визначається точність

$$\varepsilon = t_\varphi \sqrt{\frac{p(1-p)}{N}}, \tag{4.14}$$

яка обернено пропорційна $\sqrt{N}$.

З виразу (4.14) нескладно розрахувати число експериментів N, необхідних для отримання оцінки $\hat{p}$ з точністю ε і достовірністю Q:

$$N = t_\varphi^2 \frac{p(1-p)}{\varepsilon^2}. \tag{4.15}$$

Виразами (4.14), (4.15) для визначення точності ε і числа експериментів *N* можна було б користуватися, якби було відомо значення ймовірності *p*.

На практиці для визначення *p* проводиться, так зване, попереднє моделювання, яке полягає у реалізації $N = N_0$ експериментів, за результатами яких визначається значення $p_0 = \frac{m}{N_0}$, яке і приймається як значення *p*, тобто $p = p_0$.

Зауважимо, що з виразу (4.15) для числа експериментів *N* випливає істотна залежність значення *N* від дисперсії оцінюваної випадкової величини $D[\hat{p}] = \frac{1}{N} p(1-p)$. Тому, очевидно, вигідно з точки зору витрат ресурсів на проведення експериментів вибирати такі оцінювані показники ефективності модельованої системи *S*, які мають малі значення дисперсії, що призводить до незначного числа необхідних експериментів.

Правило автоматичної зупинки експерименту реалізується в підпрограмі (ПП) закінчення імітації. Це правило включається в зазначену ПП зазвичай двома способами:

- на основі виразу (4.15);
- на основі припущення про розподіл імовірностей вихідних характеристик.

Перший спосіб реалізується шляхом дворазового прогону, коли спочатку виконується прогін з N_0 експериментів, що дозволяє отримати значення $p_0 = \frac{m}{N_0}$, яке прирівнюється до значення *p*, тобто $p = p_0$, у виразі (4.15). Після цього визначається *N* за допомогою (4.15), яке і використовується як критерій зупинки експериментів. При цьому, якщо $N_0 \ge N$, то експерименти завершуються відразу, а якщо $N_0 < N$, то необхідно виконати другий прогін, реалізувавши $N - N_0$ експериментів, і завершити виконання експериментів.

Другий спосіб полягає в реалізації послідовного аналізу для визначення мінімально необхідного числа експериментів *N*. Практика показує, що використання цього способу дозволяє одержати таке число експериментів *N*, яке часто у два рази менше числа *N*, одержуваного на основі першого способу, що призводить до явного скорочення використовуваних ресурсів.

Відповідно до другого способу число *N* розглядається як випадкова величина, що залежить від результатів ($N-1$) попередніх експериментів.

У послідовному аналізі обсяг вибірки, тобто значення *N*, не фіксується. Після чергового *i*-го експерименту приймається одне з можливих рішень:

- прийняти деяку гіпотезу;
- прийняти альтернативну гіпотезу;
- продовжити експеримент, тобто провести наступний $(i+1)$ експеримент.

Таке послідовне планування процесу проведення експериментів ґрунтується на принципі максимальної правдоподібності і послідовній перевірці статистичних гіпотез, сутність чого полягає в наступному.

Передбачається, що розподіл генеральної сукупності характеризується функцією густини ймовірностей $Y = f(y,\theta)$ з невідомим параметром θ.

Висуваються дві гіпотези – H_0 і H_1. Гіпотеза H_0 полягає в тому, що значення невідомого параметра θ дорівнює θ_0, тобто $\theta = \theta_0$. Інша гіпотеза H_1 полягає в тому, що $\theta = \theta_1$.

Висунуті гіпотези перевіряються на підставі вибірки наростаючого об’єму m.

Ймовірність отримання вибірки об’ємом m за умови, що вірна гіпотеза H_0, дорівнює

$$P_{0m} = f(y_1,\theta_0)\cdot f(y_2,\theta_0)\cdot\ldots\cdot f(y_m,\theta_0).$$

Ця ж ймовірність, але за умови, що вірна гіпотеза H_1, визначається як

$$P_{1m} = f(y_1,\theta_1)\cdot f(y_2,\theta_1)\cdot\ldots\cdot f(y_m,\theta_1).$$

Процедура перевірки правильності висунутих гіпотез реалізується на основі відношення правдоподібності $\dfrac{P_{1m}}{P_{0m}}$.

Після кожного чергового експерименту визначаються P_{1m} та P_{0m} з подальшою перевіркою умов і прийняття рішення:

- якщо $\dfrac{P_{1m}}{P_{0m}} \le B$, то приймається гіпотеза H_0;
- якщо $\dfrac{P_{1m}}{P_{0m}} < A$ або $\dfrac{P_{1m}}{P_{0m}} > B$, то експерименти продовжуються;
- якщо $\dfrac{P_{1m}}{P_{0m}} \ge A$, то приймається гіпотеза H_1.

У зазначених умовах $0 < B < 1,\ A > 1,\ m \in \overline{1,N}$.

Для забезпечення збіжності необхідно, щоб

$$A \le \frac{1-\beta}{\alpha};\ B \ge \frac{\beta}{1-\alpha},$$

де α – ймовірність помилки першого роду, тобто це ймовірність відкинути гіпотезу H_0, якщо вона вірна; β – ймовірність помилки другого роду, тобто це ймовірність прийняти гіпотезу H_1, якщо вона помилкова.

Наприклад, нехай випадкова величина y підпорядковується нормальному закону розподілу з відомою дисперсією σ^2 і невідомим математичним очікуванням μ. Висуваються дві гіпотези H_0, H_1. Гіпотеза H_0 полягає в тому, що $\mu = \mu_0$, а гіпотеза H_1 – в тому, що $\mu = \mu_1$.

Оскільки

$$P_{1m} = \frac{e^{-\frac{1}{2\sigma_2}\sum\limits_{i\in\overline{1,m}}(y_i-\mu_1)^2}}{(\sqrt{2\pi\sigma})^m}; \quad P_{0m} = \frac{e^{-\frac{1}{2\sigma_2}\sum\limits_{i\in\overline{1,m}}(y_i-\mu_0)^2}}{(\sqrt{2\pi\sigma})^m},$$

то $\dfrac{P_{1m}}{P_{0m}} = e^{-\frac{1}{2\sigma_2}\left(\sum\limits_{i\in\overline{1,m}}(y_i-\mu_1)^2 - \sum\limits_{i\in\overline{1,m}}(y_i-\mu_0)^2\right)}$.

Після отримання зазначеного відношення правдоподібності для розглянутого випадку перевіряються умови і приймаються ті чи інші рішення.

4.6. Обробка та аналіз результатів експериментів з моделями систем

Після складання плану експериментів з моделлю системи, для реалізації цього плану необхідно зібрати результати експериментів у вигляді певних статистик моделювання і організувати їх оброблення відповідно до мети моделювання. Оброблені результати подаються в заданому вигляді досліднику або особі, що приймає рішення.

Як вже зазначалося в підрозділі 4.3, збирання статистик моделювання реалізується відповідною підпрограмою імітаційної моделі, а оброблення результатів експериментів та їх видача досліднику проводиться підпрограмою закінчення імітації.

Вибір методів оброблення здійснюється з обов'язковим урахуванням таких особливостей експериментів з моделями систем [6].

По-перше, експерименти з моделями систем дозволяють отримувати вибірки досить великого обсягу для кількісної оцінки характеристик процесу функціонування досліджуваної системи. Такі великі обсяги вибірок дозволяють отримувати оцінки параметрів високої точності і достовірності. Однак при цьому виникає проблема зберігання проміжних результатів у процесі оброблення великих масивів інформації.

Ця проблема, як правило, розв'язується шляхом розроблення рекурентних алгоритмів обробки інформації, що дозволяє отримувати оцінки у процесі проведення експериментів.

По-друге, складність модельованих систем не дозволяє апріорі висловити досить обґрунтоване судження про закон розподілу, наприклад, вихідних характеристик системи. Тому широко використовується практика оцінення моментів розподілу – математичного очікування, дисперсії, кореляційного моменту та ін.

По-третє, блочність конструкції моделі складної системи зумовлює можливість роздільного дослідження окремих частин. Таке дослідження можливе за умови програмної імітації вхідних змінних для однієї з частин моделі за оцінками вихідних змінних іншої частини.

До якості оцінок параметрів, одержуваних у результаті оброблення експериментів, пред'являються відомі з теорії статистики вимоги [19]:

- незсуненість оцінки, коли $M[\hat{x}] = x$, де $\hat{x}$ – оцінка параметра x;
- ефективність оцінки, коли оцінка $\hat{x}$ така, що забезпечується мінімум дисперсії, тобто $\min M\left[(\hat{x}-x)^2\right]$;
- узгодженість оцінки, коли забезпечується збіжність за ймовірністю оцінки $\hat{x}$ до оцінюваного параметра x при збільшенні числа експериментів, тобто

$$\lim_{N\to\infty} P\left[|\hat{x}-x| \ge \varepsilon\right] = 0,$$

де ε – величина похибки.

У тому випадку, коли метою експериментів є отримання саме закону розподілу, а не моментів розподілу, то за результатами експериментів отримують значення вибіркового закону розподілу $\hat{F}(x)$ або функції густини $\hat{f}(x)$. Потім висувається гіпотеза про відповідність одержаного емпіричного розподілу якому-небудь теоретичному розподілу. Перевірка гіпотези здійснюється за допомогою статистичних критеріїв узгодженості Колмогорова, Пірсона, Смірнова, Стьюдента та ін. [19].

Статистики моделювання у вигляді множини значень параметрів і їх подальше оброблення методами статистики дозволяють провести аналіз зв'язків між величинами параметрів. Для цього використовуються методи кореляційного, регресійного і дисперсного аналізу [19].

За допомогою *кореляційного аналізу* визначають, наскільки тісним є зв'язок між випадковими величинами. Тіснота зв'язку встановлюється за допомогою коефіцієнтів кореляції

$$r_{\xi\eta} = \frac{M[\xi - M[\xi]] \cdot M[\eta - M[\eta]]}{\sqrt{D[\xi]D[\eta]}},$$

де $M[\xi]$, $M[\eta]$ – математичні очікування випадкових величин ξ і η; $D[\xi]$, $D[\eta]$ – дисперсії випадкових величин ξ і η.

За допомогою статистик моделювання або, інакше, на основі N реалізацій експериментів визначається оцінка коефіцієнта кореляції:

$$\hat{r}_{\xi\eta} = \frac{\sum\limits_{k=\overline{1,N}} (x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum\limits_{k=\overline{1,N}} (x_k - \bar{x})^2 \sum\limits_{k=\overline{1,N}} (y_k - \bar{y})^2}},$$

де $\bar{x} = \frac{1}{N} \sum\limits_{k\in\overline{1,N}} x_k$, $\bar{y} = \frac{1}{N} \sum\limits_{k\in\overline{1,N}} y_k$.

Значення $\hat{r}_{\xi\eta}$ характеризує близькість залежності між випадковими величинами ξ і η до лінійної. При $|r_{\xi\eta}| = 1$ має місце функціональна, нестохастична, лінійна залежність виду

$$y = b_0 + b_1 x.$$

При $|r_{\xi\eta}| = 0$ має місце взаємна некорельованість.

Зауважимо, що незалежні випадкові величини завжди некорельовані. Однак некорельовані випадкові величини можуть бути залежними.

Випадок, коли $0 < r_{\xi\eta} < 1$ відповідає або наявності лінійної кореляції з розсіюванням, або наявності нелінійної кореляції результатів експериментів.

Як зазначалося, кореляційний аналіз дозволяє встановити існування зв'язку та його ступінь, але при цьому не встановлюється модель зв'язку між параметрами або змінними, які цікавлять дослідника.

Для визначення моделі зв'язку використовується *регресійний аналіз*. У результаті проведення експериментів отримують набори даних у вигляді певних значень параметрів, що цікавлять дослідника. Найчастіше це вихідні параметри з одного боку, а з іншого – це вхідні або внутрішні параметри системи.

Візуальний аналіз указаних даних дозволяє висловити припущення про структуру моделі або, інакше кажучи, визначити вид моделі. Після цього обчислюються значення параметрів моделі, які б мінімізували похибку між даними, отриманими на основі експериментів і моделі. Дана задача часто розв'язується за допомогою методу найменших квадратів [19].

При обробленні та аналізі результатів експериментів виникає задача порівняння середніх вибірок. Наприклад, нехай проведено n серій експериментів, у результаті яких отримані n сукупностей випадкової змінної y – $\{y^{(1)}\}$, $\{y^{(2)}\}$, …, $\{y^{(n)}\}$. Якщо виявиться, що математичні очікування зазначених сукупностей $M[y^{(1)}]$, $M[y^{(2)}]$, …, $M[y^{(n)}]$

відрізняються незначно, то сукупність $\{y^{(1)}\}$, $\{y^{(2)}\}$, …, $\{y^{(n)}\}$утворює однорідний статистичний матеріал, який можна розглядати як об'єднану сукупність вигляду $\{y^{(1)}, y^{(2)},..., y^{(n)}\}$, що значно збільшує обсяг інформації про властивості об'єкта, який вивчається на основі моделі.

Перевірку гіпотези про рівність математичних очікувань можна реалізувати за допомогою одного з критеріїв узгодженості (Смірнова, Стьюдента чи іншого) та процедури попарного порівняння математичних очікувань сукупностей. При великому числі сукупностей число попарних порівнянь буде значним, що робить дану перевірку гіпотези про рівність неефективною.

Набагато ефективніша перевірка гіпотези про рівність математичних сподівань реалізується на основі дисперсійного аналізу, який полягає у перевірці гіпотези про тотожність вибіркової і генеральної дисперсій.

4.7. Мови імітаційного моделювання

Як зазначалося, на етапі розроблення програмної реалізації імітаційної моделі важливою дією є вибір мови, за допомогою якої створюється програмний продукт.

Мови імітаційного моделювання найчастіше відносять до числа процедурно-орієнтованих, тобто до таких, які призначаються для розв'язання певного класу задач і тому містять інструкції, зручні для формулювання способів розв'язання типових задач заданого класу.

За останні роки розроблено та модернізовано велику кількість мов. Для того щоб орієнтуватися в цій множині, пропонуються різні класифікації. Однією з них мови імітаційного моделювання поділяються на класи за типами процесів, які моделюються. У результаті виділяється клас мов неперервного моделювання (continuous system simulation languages), до якого належать MIMIC, DYNAMO та ін. Далі виділяється клас мов дискретного моделювання (discrete event simulation), відомими представниками якого є GPSS, SIMULA, SIMSCRIPT та ін. Ще одним класом є клас мов комбінованого або дискретно-неперервного моделювання (combined discrete continuous simulation languages), представниками якого є GASP, FORSIM та ін.

Всі ці мови моделювання об'єднує те, що в них обов'язково передбачається набір таких програмних засобів і понять, які забезпечують вирішення наступних шести проблем.

Першою є проблема *суміщення*, яка полягає в тому, що процеси в модельованих системах, які протікають паралельно, повинні відображатися послідовно функціонуючим комп'ютером. Ця проблема, як зазначалося, вирішується шляхом введення модельного або системного часу. При цьому використовуються в основному два методи: метод фіксованого інтервалу (кроку) і метод змінного інтервалу або кроку до наступної події, детально описані в підрозділі 4.3.

Друга – це проблема *розміру*. Більшість модельованих систем мають складну структуру і алгоритми поведінки, і тому їх моделі великі за обсягом, тобто для їх розміщення потрібен великий обсяг пам’яті комп’ютера. У зв’язку з цим використовується динамічний розподіл пам’яті, коли необхідні компоненти моделі з’являються в оперативній пам’яті або залишають її залежно від поточного стану. Динамічний розподіл пам’яті можливий у випадку, коли модель побудована за блоковим (модульним) принципом.

Третьою є проблема *змін*. Для динамічних систем характерна постійна зміна стану. Тому в мовах моделювання передбачається оброблення списків, що відображають зміну стану процесів функціонування модельованої системи.

Четвертою є проблема *взаємопов’язаності*. Виникнення подій в моделі функціонування системи зумовлюється досить складними умовами через наявність великої кількості взаємних зв’язків між компонентами моделі. Для виразу цих умов необхідні засоби алгебри, логіки і теорії множин.

П’ята – це проблема *стохастичності*. У моделях складних систем виникає необхідність завдання випадкових впливів у формі простих і складних випадкових подій, випадкових змінних, функцій, процесів. У мовах моделювання передбачається можливість їх отримання на основі програм генерації послідовностей псевдовипадкових, квазірівномірно розподілених чисел в інтервалі (0,1) з наступним перетворенням цих чисел в послідовність випадкових величин із заданим законом розподілу.

Шостою є проблема *аналізу*, яка вирішується шляхом розроблення засобів для аналізу чутливості моделі до варіацій її параметрів.

4.8. Система моделювання GPSS

Основними елементами GPSS-моделі є динамічні і статистичні об’єкти [25].

До динамічних об’єктів відносять транзакти, які є аналогами динамічних елементів модельованої системи. Зокрема, якщо модельованою системою є магазин, то транзактами є покупці, якщо модельованою системою є виробнича дільниця, то транзактами можуть бути заготовки майбутніх деталей, і т. д.

До статичних об’єктів належать насамперед блоки, які являють собою підпрограми на макроасемблері або мовою C з параметрами (операндами) для звернення до них.

Внутрішній механізм передачі управління реалізується за допомогою руху транзактів від блока до блока в модельному часі. Нехай рух визначається блок-схемою.

У момент входу транзакта в блок викликається відповідна підпрограма, що реалізує функцію блока. Переміщення транзакта продовжується до тих пір, поки не виконається одна з таких можливих умов:

- транзакт входить у блок, що виконує функцію його затримки на певний час;
- транзакт входить у блок, що виконує функцію його видалення з моделі;
- транзакт намагається увійти в наступний блок, який його не приймає і тому транзакт залишається в попередньому блоці до тих пір, поки не зміняться умови моделі так, що спроба увійти в наступний блок виявиться успішною.

Об’єктами в GPSS є об’єкти типу "ресурс", "черга" і "таблиця".

Об’єкти типу "ресурс" є аналогами обслуговуючих пристроїв систем масового обслуговування, які можуть бути одноканальними, багатоканальними обслуговуючими пристроями (ОКП, БКП) і логічними ключами.

ОКП, за умови що він вільний, може бути зайнятий тільки одним транзактом у даний момент часу. БКП являє собою кілька паралельно функціонуючих ОКП і тому може бути використаний для одночасного обслуговування декількох транзактів.

У системах масового обслуговування можуть виникати ситуації, коли необхідно перервати на деякий час роботу обслуговуючих пристроїв. Ця необхідність може бути пов’язана з проведенням профілактики, ремонту та інших видів робіт, що забезпечують нормальне функціонування обслуговуючих пристроїв. Такі події можуть заблокувати або змінити рух транзактів.

Для моделювання зазначеної ситуації вводяться логічні ключі, які транзактом можуть переводитися в положення "включено" або "виключено". Ці положення логічного ключа використовуються іншими транзактами для вибору напрямку руху або очікування моменту зміни стану ключа.

Рух потоку транзактів може бути затримано через відсутність вільних ОКП або БКП. У результаті чого перед зайнятими обслуговуючими пристроями виникають черги, які в GPSS відображаються відповідними об’єктами типу "черга".

Нарешті, об’єкт "таблиця" використовується для збирання статистики випадкових величин, що характеризують рух транзактів і їх процес обслуговування.

Об’єкти типу "ресурс", "черга" і "таблиця" мають властивості і методи, які змінюють ці властивості. У GPSS ці властивості називають стандартними числовими атрибутами (СЧА), для виокремлення яких використовується інтерпретатор.

У об’єкті типу "ресурс" інтерпретатор забезпечує автоматичний розрахунок таких СЧА, як загальний час зайнятості, кількість транзактів, які займали пристрій, коефіцієнт використання пристрою, середній час зайнятості пристрою одним транзактом та ін.

В об’єкті типу "черга" інтерпретатор розраховує довжину черги, середній час перебування у черзі й інші. Інтерпретатором забезпечується певна дисципліна обслуговування, прикладами якої є черги типу FIFO ("First in, first out" – "перший прийшов, перший обслужився") і черга типу LIFO ("Last in, first out" – "останній прийшов, перший обслужився") та інші. Для реалізації дисциплін обслуговування використовуються списки користувачів.

У об’єктах типу "таблиця" інтерпретатор забезпечує розрахунок математичного очікування, середньоквадратичного відхилення та інших числових характеристик.

Загалом програма-інтерпретатор реалізує управління процесом моделювання. Вона, крім зазначеного розрахунку СЧА, реалізує:

- запуск процесу моделювання;
- стеження за рухом кожного транзакта, викликаючи по ходу руху необхідні програми обслуговування;
- ведення списку упорядкованих у часі подій, що зветься списком майбутніх подій (СМП);
- зміну модельного часу шляхом просування змінної ГОДИННИК (CLOCK) від моменту виникнення однієї події до моменту виникнення найближчої події або подій.

Всі зазначені дії виконуються за допомогою операторів GPSS, які поділяють на блоки і команди. Формат GPSS-блока визначається записом

<мітка> <операція> <операнди> <; коментарі>

Міткою визначається ім’я блока у вигляді послідовності (до 250) символів, що починається з букви латинського алфавіту. Крім букв латинського алфавіту допускається використання цифр і знака підкреслення.

Операціями блоків є дієслова, що виражають функції блоків. Наприклад, GENERATE (ГЕНЕРУВАТИ), TERMINATE (ЗАВЕРШИТИ) і т.п.

Операнди блоків є характеристиками функцій, виконуваних блоками. Число операндів не перевищує семи і вони позначаються символами A, B, C, D, E, F, G. Операнди йдуть один за одним і відокремлюються комами або одним пробілом. За відсутності операнда замість нього записується кома.

Коментарі служать для додаткових пояснень і не є обов'язковими. Коментарі відокремлюються від поля операндів символом «;».

Команди необхідні для програмування моделі і для інтерактивної взаємодії з моделлю. Зазвичай до команд відносять оператори опису даних, оператори управління та команди мови GPSS World.

Запуск процесу моделювання реалізується за допомогою команди START, яка має формат

START A, [B], [C], [D].

Зауважимо відразу, що при описі форматів квадратні дужки [] означають необов'язковість відповідного поля.

Операнд A є цілим позитивним числом, яке визначає момент закінчення прогону моделі. Зазвичай цим числом є поточне значення лічильника завершення, яке визначається числом транзактів, що обслуговуються блоками. Для зменшення значення лічильника використовується блок TERMINATE, який видаляє транзакти з моделі. Очевидно, що моделювання завершується, коли A приймає значення нуль, тобто коли з моделі будуть видалені всі транзакти.

Операнд B використовується для виведення статистики. Типово, коли операнду не присвоюється жодне значення, виводиться стандартна статистика. Коли операнду присвоюється значення «NP», статистика не виводиться. Операнд C не використовується в сучасних версіях GPSS. Операнд D використовується для виведення списку поточних подій (СПП), що включає події на поточний момент модельного часу, і вже згадуваного списку майбутніх подій (СМП). Якщо операнду присвоюється деяке ціле позитивне значення, то списки виводяться. Якщо операнду присвоюється значення нуль, то списки не виводяться.

Після виконання команди START інтерпретатор або програма управління моделюванням (ПУМ) звертається до всіх блоків GENERATE, які використовуються для створення транзактів і введення їх у модель.

Формат блока визначається записом

GENERATE [A], [B], [C], [D], [E],

де A – середнє значення інтервалу надходження транзактів у модель; B – величина розкиду можливих значень відносно середнього значення; C –

момент часу, в який з’являється перший транзакт; D – обмежувач загального числа транзактів, яке вводиться в модель через даний блок GENERATE; E – рівень або клас кожного з транзактів, що задається за допомогою чисел від 0 до 127.

Наприклад,

GENERATE 4, 2.

Такий формат блока відповідає завданню рівномірного закону розподілу інтервалів між моментами надходження транзактів з середнім значенням A = 4 і величиною розкиду можливих значень відносно середнього значення B = 2. Таким чином, можливими значеннями інтервалу будуть:

2, 3, 4, 5, 6.

Моменти появи транзактів заносяться в СМП, після чого ПУМ звертається до СПП. Оскільки СПП спочатку порожній, то ПУМ звертається до СМП з метою вибору транзактів і мінімальним моментом появи і перенесення їх у СПП. Перший транзакт з СПП просувається по блоках моделі. У тому випадку, коли відбувається затримка переміщення транзакта з причини, не пов’язаної з функціонуванням блока ADVANCE, то він залишається в СПП і ПУМ переміщує такий транзакт з цього списку далі по блоках. При попаданні транзакта у блок ADVANCE планується його вихід, що і фіксується в СМП.

У блоці ADVANCE реалізується затримка транзакта на час його обслуговування відповідним пристроєм у модельованій системі. Інтервал часу обслуговування є випадковою величиною, яка описується заданим законом розподілу.

Формат блока визначається записом ADVANCE A [B], де A – середній час затримки на обслуговування; B – величина розкиду можливих значень відносно середнього за умови їх рівномірного розподілу.

Наприклад, якщо ADVANCE 5.3, то це означає, що час обслуговування є випадковою рівномірно розподіленою величиною в інтервалі [2.8].

Пристрої обслуговування в системі зазвичай мають імена у вигляді набору символів або числа. Процес обслуговування передбачає зазначення імен тих пристроїв, в яких починається і завершується обслуговування. Для цього використовуються блоки SEIZE (зайняти) і RELEASE (звільнити), які мають такі формати:

SEIZE A,

де A – символьне або числове ім’я зайнятого пристрою, і

RELEASE A,

де A – символьне або числове ім’я пристрою, що вивільняється.

Блоки SEIZE і RELEASE, як правило, використовуються спільно з блоком ADVANCE, наприклад:

SEIZE OBS
ADVANCE 5,2
RELEASE OBS.

Згідно з прикладом транзакт забере обслуговуючий пристрій з ім’ям OBS, буде обслуговуватися протягом одного з можливих інтервалів часу {3, 4, 5, 6, 7}, звільнить обслуговуючий пристрій з ім’ям OBS. Після всього цього інтерпретатор робить спробу переміщення транзакта в наступний блок моделі.

Блоки QUEUE і DEPART у системі моделювання GPSS забезпечують можливість обліку інформації про очікування транзактів в об’єктах «черга». До цієї інформації належить:

- число входів транзактів ув чергу;
- число транзактів, які мають нульовий час очікування;
- максимальна довжина черги;
- середнє число транзактів, що знаходилися в черзі;
- середній час очікування транзактів у черзі.

Оскільки об’єктів типу «черга» може бути декілька, то для їх розрізнення виникає необхідність присвоєння імен реєстраторам цих черг, які вводяться в модель за допомогою пари блоків:

QUEUE A [B],
DEPART A [B],

де операнд A означає ім’я черги, в яку необхідно стати транзакту або яку необхідно залишити. Ім’я може бути числовим або символічним.

Операнд B визначає число транзактів, на яке збільшується або зменшується черга.

Вхід транзакта у блок QUEUE викликає виконання наступних дій:

- показання лічильника входів і значення довжини черги збільшуються на B;
- значення поточної довжини черги заноситься у стандартний числовий атрибут QS (ім’я черги);
- транзакт приєднується до черги із запам’ятовуванням її імені і значенням поточного модельного часу.

При вході транзакта у блок DEPART він залишає чергу, а інтерпретатор виконує наступні дії:

- довжина черги зменшується на B;
- якщо транзакт перебував у черзі нульовий час, то він визначається як транзакт з нульовим перебуванням з подальшою зміною показників лічильника нульових входжень;

- знищується «прив’язка» транзакта до черги.

Наявність у моделі об’єктів типу «черга» приводить до реалізації інтерпретатором розрахунку наступних статистичних даних:

- показання лічильника входів;
- максимальне і середнє значення довжини черги;
- поточне значення довжини черги (наприкінці періоду моделювання);
- середній час знаходження транзакта в черзі.

Можуть бути розраховані й інші дані.

Нехай, наприклад, є обслуговуючий пристрій USTR, перед яким можливе виникнення черги. Якщо необхідно отримати статистичні дані про очікування в черзі, то це можна реалізувати за допомогою наступного фрагмента моделі:

```
QUEUE QUSTR
SEIZE USTR
DEPART QUSTR
ADVANCE 5, 3
RELEASE USTR
```

Очевидно, що в даному прикладі всі транзакти проходитимуть пару блоків QUEUE-DEPART навіть у разі, коли пристрій USTR вільний.

Для забезпечення переходу транзакта у блок, відмінний від наступного, використовується блок TRANSFER. Основними є три режими використання блока TRANSFER.

У режимі безумовної передачі блок TRANSFER має наступний формат:

TRANSFER, B.

Операнд B визначає позицію блока, в яку повинен перейти транзакт. Позиція блока задається номером або міткою блока.

У статистичному режимі передача транзакта в один з двох блоків реалізується випадковим чином. У цьому режимі формат блока TRANSFER задається наступним записом:

TRANSFER A, [У], З

Операнд A являє собою значення ймовірності передачі транзакта у блок, позиція якого задається операндом C.

Операнд B задає позицію блока, в яку переходить транзакт з імовірністю 1-A.

Значення ймовірності задається з точністю до тисячних часток. Наприклад,

TRANSFER .435, MUSTR 1, MUSTR 2

З імовірністю 0.435 транзакт перейде у блок з міткою MUSTR 2 і з імовірністю 0,565 – у блок з міткою MUSTR 1.

Режим BOTH визначається записом

TRANSFER BOTH, B, C.

У цьому режимі транзакт спочатку прагне потрапити у блок, який вказується операндом B. Якщо це не вдається, то транзакт направляється у блок, який вказується операндом C. Якщо транзакту не вдається потрапити ні у блок, зазначений операндом B, ні у блок. вказаний операндом C, то він залишається у блоці TRANSFER до наступного перегляду списку поточних подій і буде повторювати спроби переходу в зазначеному порядку.

Усі розглянуті блоки використовуються зазвичай для моделювання одноканальних СМО. Для моделювання багатоканальних СМО використовують ще додаткові блоки, опис яких є у відповідній літературі [21]. Однак і опису розглянутих блоків достатньо для розуміння принципів роботи системи моделювання GPSS.

Висновки

Дана глава присвячена питанням імітаційного моделювання. Розглянуто основні випадки його застосування, а також переваги і недоліки імітаційного моделювання, пов’язані з особливостями побудови імітаційних моделей. Проаналізовано різні способи реалізації модельного часу. Виділено структури імітаційного моделювання, які реалізуються методами процесної взаємодії, планування подій, сканування дій і трьохфазовим методом. Описана структурна схема імітаційної моделі. Детально описані технологічні етапи процесу побудови імітаційної моделі, які включають визначення цілей моделювання, розробку концептуальної моделі, програмну реалізацію імітаційної моделі, перевірку здійсненності вимог до моделі; планування і проведення експериментів, а також аналіз результатів моделювання. Особливу увагу приділено формалізації процесу планування експериментів та аналізу отриманих результатів. Для програмної реалізації імітаційної моделі пропонується користуватися системою моделювання GPSS.

Завдання

1. Сформулюйте цілі моделювання, опишіть досліджуваний процес і побудуйте імітаційну модель згідно з наведеним описом. Обґрунтуйте своє рішення і проведіть імітацію вручну.

Припусимо, є перукарня з двома перукарями. Якщо обидва перукарі зайняті, то клієнти, що прибувають, очікують в єдиній черзі. Клієнти обслуговуються в порядку прибуття. Після обслуговування клієнт залишає перукарню. Також клієнт йде, якщо йому доводиться чекати довше 20 хв у черзі. Припустимо, перукарню відвідали 10 клієнтів. Час їх прибуття та обслуговування відомі (табл. 4.1).

Таблиця 4.1 – Час прибуття і обслуговування

Номер клієнта	Проміжок часу між прибуттям клієнтів, хв	Час обслуговування, хв
1	15,6	45,6
2	23,7	42,8
3	38,9	34,7
4	12,5	36,8
5	14,7	28,9
6	16,2	42,0
7	18,9	32,8
8	25,1	43,7
9	34,8	34,7
10	22,6	32,5

2. Сформулюйте цілі моделювання, опишіть досліджуваний процес і побудуйте імітаційну модель згідно з наведеним описом. Обґрунтуйте своє рішення і проведіть імітацію вручну.

Уявімо, що є проста телефонна система з єдиною вихідної лінією. Якщо лінія зайнята, телефонний дзвінок стає в чергу. Дзвінки обслуговуються в порядку надходження. Максимальний час очікування дзвінка становить 5 хв. Припустимо, надійшло 10 дзвінків. Час дзвінків та їх тривалість відомі (табл. 4.2).

Таблиця 4.2 – Час надходження і тривалість дзвінків

Номер дзвінка	Проміжок часу між дзвінками, хв	Тривалість дзвінка, хв
1	3,8	5,1
2	3,5	4,5
3	4,2	4,8
4	3,1	3,9
5	2,4	4,7
6	4,3	5,2
7	2,7	5,0

Закінчення таблиці 4.2

Номер дзвінка	Проміжок часу між дзвінками, хв	Тривалість дзвінка, хв
8	2,1	4,5
9	2,5	3,8
10	3,4	5,7

3. Сформулюйте цілі моделювання, опишіть досліджуваний процес і побудуйте імітаційну модель згідно з наведеним описом. Обґрунтуйте своє рішення і проведіть імітацію вручну.

Розглянемо роботу заправної станції. Припустимо, на заправній станції працює дві заправні колонки. Автомобілі обслуговуються в порядку прибуття. Припустимо, на заправну станцію прибуло 10 автомобілів. Відомий час прибуття і час, необхідний для заправки та оплати (табл. 4.3).

Таблиця 4.3 – Час прибуття і обслуговування

Номер автомобіля	Проміжок часу між прибуттям автомобілів, хв	Час обслуговування, хв
1	14,9	19,3
2	15,6	18,4
3	13,9	21,0
4	12,8	17,9
5	14,5	21,3
6	15,9	18,0
7	16,3	16,7
8	15,8	19,2
9	17,0	18,3
10	12,2	21,0

4. Сформулюйте цілі моделювання, опишіть досліджуваний процес і побудуйте імітаційну модель згідно з наведеним описом. Обґрунтуйте своє рішення і проведіть імітацію вручну.

Розглянемо роботу кавового автомата. Припустимо, покупці приходять і купують каву. Кавовому автомату необхідний деякий час для того, щоб отримати гроші, видати здачу і приготувати напій. Клієнти обслуговуються в порядку приходу. Після обслуговування клієнт йде. Припустимо, кавовий автомат обслужив 10 осіб. Час їх прибуття та обслуговування відомі (табл. 4.4).

Таблиця 4.4 – Час прибуття і обслуговування

Номер клієнта	Проміжок часу між прибуттями клієнтів, хв	Час обслуговування, хв
1	3,2	3,8
2	7,8	3,5
3	4,3	4,2
4	2,6	3,1
5	1,9	3,4
6	2,8	4,3
7	5,7	3,7
8	4,1	3,1
9	2,8	3,5
10	4,2	3,4

5. Сформулюйте цілі моделювання, опишіть досліджуваний процес і побудуйте імітаційну модель згідно з наведеним описом. Обґрунтуйте своє рішення і проведіть імітацію вручну.

Розглянемо роботу кавового автомата. Припустимо, що кава коштує 3 у.о. Кавовий автомат приймає монети номіналом 1, 2 і 5 у.о. і видає здачу. Передбачається, що клієнти прибувають і купують каву. Кавовий автомат витрачає 0,5 хв на те, щоб прийняти гроші, 0,3 хв – на те, щоб видати здачу, і 1 хв – на приготування кави. Припустимо, кавовий автомат обслужив 10 осіб. Час їх прибуття і номінал монет кожного відомі (табл. 4.5).

Таблиця 4.5 – Час прибуття і номінал монет

Номер клієнта	Час прибуття клієнтів, хв	Номінал монет, у.о.
1	15,6	1–2
2	17,2	5
3	19,1	1–1–1
4	21,3	1–1–2
5	23,6	2–2
6	24,9	1–2
7	26,2	2–1
8	27,5	5
9	29,6	2–2
10	33,0	1–1–1

6. Сформулюйте цілі моделювання, опишіть досліджуваний процес і побудуйте імітаційну модель згідно з наведеним описом. Обґрунтуйте своє рішення і проведіть імітацію вручну.

Розглянемо роботу кавового автомата. Передбачається, що клієнти прибувають і купують каву. Кавовий автомат витрачає близько 2 хвилин на те, щоб прийняти гроші, видати здачу і приготувати каву. Якщо кавовий автомат зайнятий, клієнти очікують в одній черзі. Після обслуговування клієнт йде. Максимальний час очікування клієнта в черзі становить 10 хв. Відомий час прибуття клієнтів (табл. 4.6).

Таблиця 4.6 – Час прибуття клієнтів

Номер клієнта	Проміжок часу між прибуттями клієнтів, хв
1	14,9
2	15,6
3	13,9
4	12,8
5	14,5
6	15,9
7	16,3
8	15,8
9	17,0
10	12,2

7. Сформулюйте цілі моделювання, опишіть досліджуваний процес і побудуйте імітаційну модель згідно з наведеним описом. Обґрунтуйте своє рішення і проведіть імітацію вручну.

Уявімо роботу системи управління світлофором на перехресті. Припустимо, система управління має два стани: «рух дозволено», «рух заборонено». Стан «рух дозволено» означає, що дозволяється рух по головній дорозі, а стан «рух заборонено» означає, що забороняється рух по головній дорозі, а дозволяється – по другорядній. Припустимо, що рух по головній дорозі припиняється, коли на другорядній дорозі з’являється 2 автомобілі. Припустимо, 10 автомобілів прибувало на другорядну дорогу і кожен автомобіль витрачав близько 1 хв на те, щоб проїхати перехрестя (табл. 4.7).

Таблиця 4.7 – Час прибуття автомобілів

Номер автомобіля	Час прибуття, хв
1	15,6
2	23,7
3	38,9
4	42,3
5	48,5
6	51,4
7	54,6
8	58,9
9	63,5
10	64,8

8. Сформулюйте цілі моделювання, опишіть досліджуваний процес і побудуйте імітаційну модель згідно з наведеним описом. Обґрунтуйте своє рішення і проведіть імітацію вручну.

Розглянемо роботу ліфта в триповерховому будинку. Ліфт викликається натисканням кнопки. Коли ліфт приїжджає, людина входить і вибирає поверх. Якщо ліфт зайнятий, людина чекає. Припустимо, ліфтом скористалося 10 осіб. Час та поверхи відомі (табл. 4.8). Передбачається, що людина заходить і виходить з ліфта відразу. Ліфт проїжджає один поверх за 1 хв.

Таблиця 4.8 – Час прибуття і поверхи

Номер людини	Час прибуття, хв	Поверх, з якого викликається ліфт	Необхідний поверх
1	1,2	1	2
2	35,2	1	3
3	44,3	1	2
4	56,2	2	1
5	75,3	2	1
6	97,4	3	1
7	110,2	1	3
8	134,6	3	1
9	156,2	1	2
10	186,4	2	1

9. Сформулюйте цілі моделювання, опишіть досліджуваний процес і побудуйте імітаційну модель згідно з наведеним описом. Обґрунтуйте своє рішення і проведіть імітацію вручну.

Розглянемо роботу світлофора на перехресті. Світлофор може перебувати в одному з трьох станів: «рух припинено», «увага» і «рух дозволено». Якщо світлофор знаходиться в стані «рух припинено», ніхто не переходить перехрестя. Якщо пішоходу необхідно перейти дорогу, він натискає кнопку, і світлофор змінює стан. Спочатку він переходить у стан «увага» і знаходиться в ньому 1 хв. Потім світлофор переходить у стан «рух припинено» на 1,5 хв і повертається в стан «рух дозволено». Припустимо, дорогу на перехресті перейшли 10 пішоходів. Час їх прибуття відомо (табл. 4.9).

Таблиця 4.9 – Час прибуття пішоходів

Номер пішохода	Час прибуття, хв
1	1,5
2	3,9
3	6,0
4	8,4
5	11,2
6	14,2
7	17,2
8	20,1
9	23,4
10	28,0

10. Час між послідовними прибуттями покупців у магазин рівномірно розподіляється в інтервалі від 1 до 20 хв. Для 50 % покупців час обслуговування становить 8 хв, а для інших 50 % – 14 хв. Запропонуйте відповідні для цього випадку 2 генератори випадкових чисел: для отримання моментів часу прибуття покупців і часу обслуговування. Зімітуйте роботу магазину протягом чотирьох годин. Визначте загальний час очікування покупців і час простою системи.

11. Розробіть алгоритм, який імітує гру в кубики згідно з такими правилами. Два кубики з числами від 1 до 6 на сторонах перемішуються і кидаються одночасно. Якщо сума чисел на верхній стороні дорівнює 2 або 12, гравець отримує 250 балів. Якщо сума дорівнює 3 або 11, гравець отримує 100 балів. Якщо сума дорівнює 4 або 10, гравець отримує 50 балів. Якщо сума дорівнює від 5 до 9, гравець втрачає 50 балів. Розрахуйте очікувані бали перемоги чи поразки.

12. Припустимо, що кожен день людина грає у гру «Трійка». Згідно з правилами гравець підкидає монету доти, поки різниця між числом орлів і решок буде дорівнювати трьом. За кожне підкидання монетки гравець

платить 1 грн, проте у разі успішної гри він отримує 5 грн. Кожен день гравець витрачає 10 грн. на цю гру. Він грає, поки не витратить усі виділені гроші або виграє. За допомогою імітаційної моделі визначте результат гри за тиждень (7 днів). Чи виграє людина?

Контрольні питання

1. Дайте визначення імітаційному моделюванню.
2. У яких випадках використовується імітаційне моделювання?
3. У яких прикладних областях часто застосовується імітаційне моделювання?
4. Наведіть приклади виробничих процесів, в яких можуть використовуватися імітаційні моделі.
5. Наведіть приклади систем обслуговування, в яких можуть використовуватися імітаційні моделі.
6. Які цілі імітаційного моделювання?
7. Які можна виділити проблеми в імітаційному моделюванні?
8. Опишіть переваги та недоліки імітаційного моделювання.
9. Опишіть порядок реалізації імітаційного моделювання компонентів складної системи.
10. Що таке активність імітаційної моделі?
11. Опишіть механізми зміни модельного часу.
12. Як розглядається протікання часу в імітаційному моделюванні?
13. Які можна виділити імітаційні структури?
14. Чим відрізняються метод процесної взаємодії та метод планування подій?
15. Чим відрізняються двофазний і трифазний методи?
16. Які частини виділяються у структурі імітаційної моделі?
17. Які етапи виділяють у процесі розроблення імітаційної моделі?
18. У чому полягає етап розробки концептуальної моделі?
19. У чому полягають процеси верифікації та валідації моделі?
20. У чому полягає основне завдання планування експериментів і як воно вирішується?
21. Наведіть класифікацію факторів, що розглядаються при плануванні експериментів з імітаційною моделлю.
22. Яким чином визначається число чинників?
23. Як визначається точність і достовірність оцінок, отриманих у результаті експерименту?
24. Як визначається число експериментів, необхідних для отримання оцінки з заданою точністю і достовірністю?

25. Як реалізується правило автоматичної зупинки експерименту?
26. Чим обумовлюється вибір методів обробки результатів експериментів з моделями систем?
27. Які вимоги пред'являються до якості оцінок параметрів, одержуваних при обробцленні результатів експериментів?
28. Для чого використовуються різні методи кореляційного, регресійного і дисперсного аналізу при аналізі результатів експериментів?
29. Які завдання вирішуються різними мовами моделювання?
30. Які об'єкти відносять до динамічних і статичних у системі моделювання GPSS?

5. МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

5.1. Визначення бізнес-процесу

Успішність роботи компанії на ринку залежить від багатьох чинників: діапазону пропонованих послуг, насиченості ринку, маркетингової політики, тощо. З метою підтримки конкурентоспроможності сучасні, орієнтовані на постійний розвиток компанії змушені постійно вдосконалювати свою діяльність, що вимагає розробки нових технологій і прийомів ведення бізнесу, впровадження більш ефективних методів управління та організації діяльності. Ось чому в числі інших заходів необхідно вміти використовувати методологію моделювання бізнес-процесів.

Сьогодні методи та інструменти моделювання бізнес-процесів є одночасно і серйозним напрямком наукових досліджень, і процвітаючим сектором ринку програмного забезпечення. Спектр методів моделювання, що застосовуються для опису бізнес-процесів, досить широкий: від найпростіших графічних нотацій, що використовуються для побудови блок-схем алгоритмів, і таких строгих математичних апаратів, як мережі Петрі [26], до об’єктно-орієнтованих мов моделювання, подібних UML (Unified Modeling Language) [27, 28], і спеціально розроблених для опису бізнес-систем методологій, наприклад XPDL (XML Process Definition Language) [29] і BPEL (Business Process Execution Language) [30].

Бізнес-процес являє собою систему послідовних, цілеспрямованих і регламентованих видів діяльності, в якій за допомогою керуючого впливу і за допомогою ресурсів входи процесу перетворюються у виходи, тобто результати процесу, що є цінністю для споживачів [31]. Операції виконуються різними елементами організаційної структури підприємства за певними бізнес-правилами. Під бізнес-правилами розуміють способи реалізації бізнес-функцій у рамках бізнес-процесу, а також характеристики і умови виконання бізнес-процесу [32].

Методики моделювання та аналізу бізнес-процесів є в даний час одним з найважливіших інструментів підвищення ефективності бізнесу. Використання подібних методик має своєю кінцевою метою реорганізацію бізнес-процесів і, як наслідок, збільшення виручки, скорочення витрат на виробництво продукції і послуг, підвищення якості продукції, оптимальне використання оборотного капіталу, впровадження систем автоматизації та багато іншого.

Перш ніж намагатися поліпшити діяльність підприємства, необхідно проаналізувати, як воно працює в даний час. Для аналізу необхідно знати не тільки, як працює підприємство в цілому, як воно взаємодіє із зовнішніми організаціями, замовниками та постачальниками, але і як організована діяльність на кожному робочому місці.

Бізнес-процес – це сукупність взаємопов'язаних заходів або завдань, спрямованих на створення певного продукту або послуги для споживачів.

Можна виділити три види бізнес-процесів:

1. *Управляючі* бізнес-процеси, які керують функціонуванням системи. Прикладом керуючого процесу може служити управління якістю, стратегічний менеджмент.

2. *Операційні* бізнес-процеси, які складають основний бізнес компанії і створюють основний потік доходів. Прикладами операційних бізнес-процесів є постачання, виробництво, маркетинг та продаж.

3. *Підтримуючі* бізнес-процеси, які обслуговують основний бізнес. Наприклад, бухгалтерський облік, підбір персоналу, технічна підтримка.

Відповідно до іншої класифікації, що прийшла з моделювання складних систем, виділяють наступні види моделей бізнес-процесів [32]:

- функціональні, що описують сукупність виконуваних системою функцій, їх входи і виходи;
- поведінкові, що показують, коли і/або за яких умов виконуються бізнес-функції, за допомогою таких категорій як стан системи, подія, перехід з одного стану в інший, умови переходу, послідовність подій;
- структурні, що характеризують морфологію системи – склад підсистем, їх взаємозв'язок;
- інформаційні, що відображають структури даних – їх склад і взаємозв'язок.

Бізнес-процес має наступні характеристики:

- визначеність: бізнес-процес повинен мати чітко визначені межі, входи і виходи;
- впорядкованість: бізнес-процес має складатися з дій, впорядкованих в часі і просторі;

- наявність споживача: на виході процесу повинен бути споживач;
- додаткове значення: перетворення всередині процесу має позитивно або негативно впливати на вихід, що цікавить споживача;
- вбудованість: процес не може існувати сам по собі, він повинен бути вбудований у структуру підприємства;
- суміжна функціональність: зазвичай процес може, але необов'язково мусить охоплювати кілька функцій.

Будь-який бізнес-процес починається із попиту споживача і закінчується його задоволенням. Процесно-орієнтовані організації намагаються усувати бар'єри і затримки, що виникають на стику двох різних підрозділів організації при виконанні одного бізнес-процесу.

Бізнес-процеси повинні бути організовані таким чином, щоб створювати цінність для своїх споживачів і виключати будь-які необов'язкові дії. На виході правильно побудованих бізнес-процесів збільшується цінність для споживача і рентабельність виробництва товару або послуги.

Для аналізу бізнес-процес може бути декомпонований на кілька підпроцесів, які мають власні характеристики, однак вони спрямовані на досягнення мети основного бізнес-процесу.

Організація розглядається як система бізнес-процесів, націлених на виконання кінцевого результату (виробництво продукції або послуги). В якості споживачів бізнес-процесу можуть розглядатися як внутрішні, так і зовнішні клієнти (споживачі, замовники продукції чи послуг, суміжні підрозділи). Метою формалізації бізнес-процесів є оцінення їх ефективності та результативності.

5.2. Стандарти і технології моделювання бізнес-процесів

Бізнес-моделювання – це діяльність з формування моделей організації, що включає опис об'єктів бізнесу (підрозділів, посад, ресурсів, ролей, процесів, операцій, інформаційних систем, носіїв інформації і т. д.) і зазначення зв'язків між ними. Вимоги до моделей і їх відповідний зміст визначаються цілями моделювання.

Бізнес-моделюванням також називають дисципліну і окремий підпроцес у процесі розроблення програмного забезпечення, в якому описується діяльність компанії і визначаються вимоги до системи, ті підпроцеси та операції, які підлягають автоматизації в інформаційній системі, що розробляється.

Моделлю бізнес-процесу називається його формалізований (графічний, табличний, текстовий, символьний) опис, що відображає реально існуючу або передбачувану діяльність підприємства.

Модель, як правило, містить такі відомості про бізнес-процеси:

- набір кроків – бізнес-функцій, що складають процес;
- порядок виконання бізнес-функцій;
- механізми контролю та управління в рамках бізнес-процесу;
- виконавців кожної бізнес-функції;
- вхідні документи та інформацію, що використовуються кожною бізнес-функцією;
- вихідні документи та інформацію, що генеруються кожною бізнес-функцією;
- ресурси, необхідні для виконання кожної бізнес-функції;
- документацію та умови, які регламентують виконання кожної бізнес-функції;
- параметри, що характеризують виконання бізнес-функцій і процесу в цілому.

Бізнес-моделювання здійснюється в одному з двох напрямків [33]. Першим є складання моделі бізнес-процесу AS-IS («як є»). Побудована AS-IS модель бізнес-процесу піддається критичному аналізу або обробляється спеціальним програмним забезпеченням з метою поліпшення. Другим напрямком бізнес-моделювання є побудова моделі TO-BE («як має бути»), яка відображає ефективність бізнес-процесу з точки зору обраних критеріїв.

Модель AS-IS будується в результаті ретельного обстеження підприємства. На її основі досягається консенсус між різними підрозділами по тому, «хто що робить» і «що кожний додає в процес». Модель AS-IS дозволяє з’ясувати, «що ми робимо сьогодні». Іншими словами. Модель AS-IS формалізує існуючі на підприємстві процеси, дозволяє виявити слабкі місця, надлишкові дії, нераціональне використання ресурсів. Крім цього, модель AS-IS є основою для постановки завдань автоматизації, тобто розробки та впровадження інформаційних систем.

Впровадження інформаційної системи неминуче призведе до перебудови існуючих бізнес-процесів підприємства. Аналіз функціональної моделі дозволяє зрозуміти, де знаходяться найбільш слабкі місця, в чому полягатимуть переваги нових бізнес-процесів і наскільки глибоким змінам піддасться існуюча структура організації бізнесу. Деталізація бізнес-процесів дозволяє виявити недоліки організації навіть там, де функціональність на перший погляд здається очевидною.

Ознакою неефективної діяльності можуть бути даремні, некеровані і дубльовані роботи, неефективний документообіг (потрібний документ не опиняється в потрібному місці в потрібний час), відсутність зворотних зв'язків з управлінням (на проведення роботи не робить впливу її результат) і входами (об'єкти або інформація використовуються нераціонально) і т. д.

Знайдені в моделі AS-IS недоліки можна виправити при розробленні моделі TO-BE – моделі нової організації бізнес-процесів. Модель TO-BE потрібна для оцінкення наслідків упровадження інформаційної системи та аналізу альтернативних (кращих) шляхів виконання роботи та документування того, як підприємство буде функціонувати в майбутньому.

Існують різні методи і техніки, що застосовуються для моделювання бізнес-процесів. Серед них виділяються методи структурного аналізу і проектування, об'єктно-орієнтованого аналізу і проектування, Business Process Model and Notation (BPMN), BPEL, Business Process Modeling Language (BPML), Event-driven Process Chain (EPC) та інші. Розглянемо їх більш докладно.

Структурним аналізом прийнято називати метод дослідження статичних характеристик системи шляхом виділення в ній підсистем і елементів різного рівня ієрархії і визначення відношень і зв'язків між ними. Сутність структурного підходу до розробклення моделі полягає в поділі аналізованої системи на частини – «чорні ящики», та ієрархічній організації цих «чорних ящиків». Перевага оперування «чорними ящиками» полягає в тому, що немає необхідності знати, як вони працюють, досить мати інформацію про їх входи і виходи, а також функції, які вони виконують.

У структурному аналізі і проектуванні бізнес-процесів використовуються різні моделі, які описують [34]:

- функціональну структуру системи;
- послідовність виконуваних дій;
- передачу інформації між функціональними процесами;
- відносини між даними.

Найбільш поширеними моделями перших трьох груп є:

- функціональна модель SADT (Structured Analysis and Design Technique);
- модель IDEF3 (Integration Definition for Function Modeling);
- DFD (Data Flow Diagram) - діаграма потоків даних.

Метод SADT являє собою сукупність правил і процедур, призначених для побудови функціональної моделі об'єкта будь-якої предметної області. Технологія SADT (перейменована в IDEF0) традиційно використовуються для моделювання організаційних систем (бізнес-процесів). Достоїнствами застосування моделей SADT для опису бізнес-процесів є:

- повнота опису бізнес-процесу (управління, інформаційні та матеріальні потоки, зворотні зв'язки);
- жорсткі вимоги методу, що забезпечують отримання моделей стандартного виду;
- відповідність підходу до опису процесів стандартам ISO 9000.

IDEF є досить поширеним сімейством методів моделювання організаційних систем, на основі якого в різних країнах розроблено множину різних нормативних документів. У даний час до сімейства IDEF прийнято відносити наступні методології [32, 35]:

- IDEF0 – методологія функціонального моделювання, забезпечена наочною графічною мовою. Дозволяє подати модельовану систему у вигляді набору взаємозалежних функцій. Як правило, моделювання засобами IDEF0 є першим етапом вивчення системи;
- IDEF1 – методологія моделювання інформаційних потоків усередині системи, що дозволяє відображати і аналізувати їх структуру та взаємозв'язки;
- IDEF1X (IDEF1 Extended) – методологія моделювання, яка застосовується для побудови інформаційної моделі, що становить структуру інформації, необхідної для підтримки функцій виробничої системи або середовища. IDEF1X часто використовується для моделювання реляційних баз даних, що мають відношення до даної системи;
- IDEF4 – методологія об'єктно-орієнтованого проектування. IDEF4 реалізує об'єктно-орієнтований аналіз великих систем, надаючи користувачеві графічну мову для зображення класів, діаграм успадкування, таксономії методів;
- IDEF5 – методологія онтологічного дослідження складних систем. Застосовуючи методологію IDEF5, онтологію системи можна описати за допомогою певного словника термінів і правил, на підставі яких можуть бути сформовані достовірні твердження про стан аналізованої системи в певний момент часу. На базі цих тверджень формуються висновки про подальший розвиток системи та проводиться її оптимізація.

Діаграми потоків даних DFD являють собою ієрархію функціональних процесів, пов'язаних потоками даних. Мета такого подання – продемонструвати, як кожен процес перетворює свої вхідні дані у вихідні, а також виявити відносини між цими процесами.

Модель системи визначається як ієрархія діаграм потоків даних, що описують асинхронний процес перетворення інформації від її введення в систему до видачі споживачеві. DFD з самого початку створювалися як засіб проектування інформаційних систем (тоді як SADT – як засіб моделювання систем взагалі) і мають більш багатий набір елементів, що адекватно відображають специфіку таких систем (наприклад, сховища даних є прообразами файлів або баз даних, зовнішні сутності відображають взаємодію модельованої системи із зовнішнім світом).

Розглянуті вище засоби структурного аналізу приблизно однакові з точки зору можливостей виразних засобів моделювання, тому одним з основних критеріїв вибору того чи іншого методу є ступінь володіння ним з боку консультанта або аналітика.

Концептуальною основою об’єктно-орієнтованого аналізу і проектування (ООАП) є об’єктна модель. Її основні принципи (абстрагування, інкапсуляція, модульність та ієрархія) і поняття (об’єкт, клас, атрибут, операція, інтерфейс та ін.) найбільш чітко сформульовані у Г. Буча [27, 28].

Більшість сучасних методів ООАП базуються на використанні мови UML [36]. Уніфікована мова моделювання UML (Unified Modeling Language) являє собою мову для визначення, подання, проектування та документування програмних систем, організаційно-економічних систем, технічних систем та інших систем різної природи. UML містить стандартний набір діаграм і нотацій найрізноманітніших видів. Їх можна розділити на ті, які моделюють статичну структуру системи (статичну модель), і ті, які моделюють динамічну структуру системи (динамічну модель). Статична модель фіксує сутності та структурні відносини між ними. Динамічна модель відображає те, як сутності взаємодіють для формування необхідної поведінки системи.

До діаграм структури належать:

- діаграма класів (class diagram);
- діаграма складової структури (composite structure diagram);
- діаграма компонентів (component diagram);
- діаграма розгортання (deployment diagram);
- діаграма об’єктів (object diagram);
- діаграма пакетів (package diagram);
- діаграма профілів (profile diagram).

До діаграм поведінки належать:

- діаграма діяльності (activity diagram);
- діаграма прецедентів (use-case diagram);
- діаграма станів (скінченних автоматів) (state machine diagram);

- діаграми взаємодії (interaction diagrams):
  - діаграма послідовності (sequence diagram);
  - діаграма комунікації (communication diagram);
  - діаграма огляду взаємодії (interaction overview diagram);
  - діаграма синхронізації (timing diagram).

Крім технологій ООАП, для моделювання бізнес-процесів можуть бути використані інші методи та мови, наприклад, EPC, BPMN, BPEL, BPML.

Event-driven Process Chain (EPC) – це вид блок-схем, що використовується для моделювання бізнес-процесів [37]. EPC можна використовувати для планування бізнес-ресурсів та покращення бізнес-процесів. EPC складається з наступних елементів:

- Функції. Базовими будівельними блоками є функції, відповідні до виконуваної дії (завдання).
- Події. Описують ситуацію до і/або після виконання функції. Функції з'єднані подіями. Подія може бути постумовою однієї функції і подавати передумову іншої функції.
- Логічні з'єднання. З'єднують функції та події, визначаючи таким чином потік управління. Виділяють три типи з'єднань: «і», «або», «виключаюче або».

Для моделювання бізнес-процесів організацією Business Process Management Initiative (BPMI) була спеціально розроблена нотація Business Process Model and Notation (BPMN) [38].

Специфікація BPMN описує умовні позначення для відображення бізнес-процесів у вигляді діаграм.

У рамках загальної моделі BPMN існує три типи підмоделей [39]:

- Private – внутрішні процеси певної організації;
- Abstract – являють собою взаємодію між внутрішнім бізнес-процесом й іншим процесом або учасником на рівні обміну повідомленнями, без розкриття деталей інших учасників або процесів.
- Collaboration – відображають взаємодію між двома і більше учасниками шляхом обміну повідомленнями. Фактично це два і більше абстрактних процесів на одній діаграмі.

BPMN підтримує тільки набір концепцій, необхідних для моделювання безпосередньо бізнес-процесів. Моделювання інших аспектів, таких як модель даних, організаційна структура, знаходиться поза зоною уваги BPMN. BPMN підтримує також властивості графічних об'єктів, що робить можливим генерацію виконуваного BPEL коду. BPMN безпосередньо відображається на мови виконання бізнес-процесів, такі як BPEL і BPML.

Business Process Execution Language (BPEL) – мова на основі XML для формального опису бізнес-процесів і протоколів їх взаємодії між собою. У той час як BPMN надає нотацію для моделювання, BPEL є мовою опису виконання процесів. BPEL розширює модель взаємодії веб-служб і включає в цю модель підтримку транзакцій.

Створення BPEL спочатку мало на меті:

- Визначити бізнес-процес, який взаємодіє із зовнішніми об’єктами через операції веб-служб, реалізовані за допомогою WSDL 1.1.
- Визначити бізнес-процес за допомогою мови на основі XML, а не у вигляді графічного зображення процесу.
- Визначити ключові концепції веб-служб, які можуть використовуватися всередині і зовні бізнес-процесу.
- Забезпечити режими ієрархічного і графічного контролю, а також зробити їх сполучення найбільш рівномірним.
- Надати функції перетворення даних для простих операцій з даними бізнес-процесу.
- Підтримати неявне створення та видалення об’єктів процесу в базовому механізмі моделювання життєвого циклу.
- Визначити довгострокову транзакційну модель для відновлення після збою.
- Використовувати веб-служби як засіб моделювання декомпозиції та компонування процесу.

Business Process Modeling Language (BPML) – це мова XML, створена для визначення формальної моделі, що виражає виконувані процеси, які описують усі аспекти корпоративних бізнес-процесів [40]. BPML визначає операції різного рівня складності, транзакції і компенсації, управління даними, паралелізм, обробку виключень і операційну семантику. Граматика BPML оформляється у вигляді XML-схеми, що забезпечує сталість визначень і їх обмін між гетерогенними системами та інструментами моделювання.

BPML – це багата і зріла мова, за допомогою якої можна описувати як прості, так і складні бізнес-процеси. Оскільки BPML і BPEL – це мови з блоковою структурою, то у них однаковий набір виразів і схожий синтаксис. У порівнянні з операціями, які підтримуються BPEL, можливості BPML ширші. Синтаксис BPML підтримує операції та їх типи, процеси, властивості, сигнали, розклади і нестандартні ситуації.

5.3. CASE-засоби моделювання бізнес-процесів

Безліч програмних продуктів провідних розробників ринку інформаційних технологій дозволяють створювати зрозумілі, зручні у використанні візуальні моделі бізнес-процесів. Серед найбільш популярних CASE-засобів для моделювання бізнес-процесів на основі структурного та об’єктно-орієнтованого підходів варто виділити наступні:

- Visual Paradigm;
- Sybase PowerDesigner;
- AllFusion Process Modeler;
- ER/Studio Business Architect;
- Rational Rose;
- Microsoft Office Visio.

Програмне забезпечення Visual Paradigm включає:

- Visual Paradigm for UML (VP-UML) – інструмент візуального моделювання додатків, генерування документації, вихідного коду і баз даних на уніфікованій мові моделювання UML.
- Smart Development Environment – програма, що розширює можливості середовищ розробки Visual Studio, Eclipse, NetBeans і IntelliJ IDEA діаграмами UML і ER (Entity / Relationship), функціями інжинірингу вихідного коду та ін.
- DB Visual ARCHITECT – засіб проектування баз даних, який підтримує функції створення бази даних з діаграмами відношень та діаграмами класів UML.
- Business Process Visual ARCHITECT – програмне рішення для моделювання бізнес-процесів та анімації бізнес-процесів, генерування звітів, публікації проектів у мережі. Це засіб моделювання бізнес-процесів, що підтримує стандарт BPMN 2.0, який дозволяє будувати діаграми потоків даних, а також організаційні діаграми.
- Agilian – засіб для побудови діаграм розробки програмного забезпечення.
- Teamwork Server – єдине сховище проектів програм для колективного доступу розробників.

Sybase PowerDesigner – це повнофункціональний інструментарій для створення бізнес-додатків, що включає засоби моделювання бізнес-процесів, можливості концептуального і фізичного проектування баз даних, можливості моделювання з використанням UML. Він надає централізований репозитарій для зберігання моделей і об’єктів. Основні особливості продукту включають:

- Моделювання бізнес-процесів на основі діаграм потоків управління.
- Технології моделювання даних (концептуальна і фізична модель), що базуються на ER стандарті, включаючи технології моделювання сховищ даних (схеми «зірка» і «сніжинка», багатовимірне моделювання, прив'язка до конкретного джерела даних).
- Стандартні діаграми UML.
- Requirement Model – спеціалізована модель для документування і аналізу вимог, що пред'являються до створюваної інформаційної системи.
- Сучасний, графічний інтерфейс, що настроюється, призначений для користувача, який містить:

– поліпшене керування моделями, включаючи синхронізацію об'єктів, моделей і баз даних;

– розширений, незалежний від моделі генератор звітів, який дозволяє отримати документ, включаючий в себе інформацію щодо декількох моделей.

AllFusion Process Modeler – це інструмент для моделювання, аналізу, документування та оптимізації бізнес-процесів. AllFusion Process Modeler можна використовувати для графічного зображення бізнес-процесів: схеми виконання робіт, обміну інформацією, документообігу, візуалізацію моделі бізнес-процесу. Графічне подання цієї інформації дозволяє перевести завдання управління організацією з області складного ремесла у сферу інженерних технологій. До особливостей AllFusion Process Modeler можна віднести наступні:

- підтримує три стандартні нотації: IDEF0, DFD і IDEF3;
- дозволяє підвищити ефективність бізнесу, оптимізувати будь-які процедури в компанії;
- повністю підтримує методи розрахунку собівартості за обсягом господарської діяльності (функціонально-вартісний аналіз, ABC-аналіз);
- недорогий, поширений, легкий в освоєнні засіб;
- дозволяє полегшити сертифікацію на відповідність стандартам якості ISO 9000;
- інтегрований з ERwin Data Modeler (для моделювання баз даних);
- інтегрований із засобом імітаційного моделювання Arena;
- містить власний генератор звітів;
- дозволяє ефективно маніпулювати моделями – зливати і декомпонувати їх;

- має широкий набір засобів документування моделей і проектів.

ER/Studio Business Architect – це засіб створення графічного зображення бізнесу, починаючи з основних концепцій і закінчуючи процесами, які докладно описують діяльність компанії. ER/Studio Business Architect дозволяє моделювати бізнес-процеси і дані, які використовуються ними.

До основних можливостей ER/Studio Business Architect відносять:

- моделювання бізнес-процесів з використанням стандартних елементів, таких як: послідовності, завдання, доріжки;
- підтримку високорівневого концептуального моделювання за допомогою предметних областей, бізнес-підрозділів, взаємодій та інших елементів;
- потужний інтуїтивний інтерфейс з організаційними ієрархіями для зручної організації інформації;
- використання нотації BPMN.

ER/Studio Business Architect дозволяє документувати й аналізувати наявні бізнес-процеси, спрощує їх перевірку, аналіз і вироблення рекомендацій щодо їх змін.

Rational Rose – це CASE-засіб фірми Rational Software Corporation (США), який призначений для автоматизації етапів аналізу і проектування програмного забезпечення, а також для генерації програмного коду і випуску проектної документації. Rational Rose використовує методологію ООАП, що базується на UML.

До переваг Rational Rose належать наступні:

- підтримка мови UML;
- генерація програмного коду;
- зворотне проектування існуючих систем;
- здатність інтегруватися із засобами розробки (Visual Studio);
- наявність засобів автоматичного контролю, в тому числі перевірки відповідності двох моделей;
- зручний для користувача графічний інтерфейс;
- багатоплатформність.

Microsoft Office Visio – це універсальний засіб моделювання даних і додатків, що підтримує створення моделей даних і об’єктно-орієнтоване моделювання додатків. Visio підтримує великий набір шаблонів: блок-схеми бізнес-процесів, схеми мереж, діаграми робочих процесів, моделі баз даних і діаграми ПЗ. Їх можна використовувати для візуалізації та раціоналізації бізнес-процесів, відстеження ходу роботи над проектами та використання ресурсів, оптимізації систем, складання схеми організаційної структури.

BPMN 2.0 Modeler для Visio є автономним розширенням Visio, яке дозволяє створювати моделі бізнес-процесів. BPMN 2.0 Modeler підтримує повний набір елементів нотації BPMN 2.0.

Таким чином, існуючі CASE-засоби дозволяють моделювати бізнес-процеси за допомогою різних нотацій і стандартів. На вибір засобу може вплинути використовуваний метод моделювання, вартість ліцензії на програмний продукт, його здатність інтегруватися з іншими засобами аналізу і проектування, а також особисті переваги аналітика.

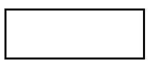
5.4. Методологія структурного аналізу і проектування бізнес-процесів

5.4.1. МЕТОДОЛОГІЯ ФУНКЦІОНАЛЬНОГО МОДЕЛЮВАННЯ IDEF0

IDEF0 – це технологія опису системи в цілому як множини взаємозалежних дій, або функцій. Важливо відзначити функціональну спрямованість IDEF0 – функції системи досліджуються незалежно від об’єктів, що забезпечують їх виконання. «Функціональна» точка зору дозволяє чітко відокремити аспекти призначення системи від аспектів її фізичної реалізації.

IDEF0 поєднує в собі невелику за обсягом графічну нотацію (вона містить тільки два позначення: блоки і стрілки) зі строгими і чітко визначеними рекомендаціями, в сукупності призначеними для побудови якісної та зрозумілої моделі системи (табл. 5.1).

Таблиця 5.1 – Елементи IDEF0 діаграми

Позначення	Назва	Опис
	Функція (Function або Activity Box)	Дія, що виконується в рамках бізнес-процесу
	Стрілка (Arrow)	Являє вхід, вихід, управління або механізм функціонального блока
[ ]	Тунель (Arrow Tunnel)	Позначення тунелю навколо початку стрілки позначає, що ця стрілка не була успадкована від функціонального батьківського блоку і з’явилася на даній діаграмі. Позначення тунелю біля кінця стрілки поблизу від блока-приймача означає, що в дочірній відносно даного блока діаграмі ця стрілка відображатися та розглядатися не буде

Модельований бізнес-процес в нотації IDEF0 показується як множина ієрархічно впорядкованих бізнес-функцій (або дій).

Кожна функція обробляє або переводить вхідні параметри (сировину, інформацію тощо) у вихідні. Оскільки моделі IDEF0 зображають систему як множину ієрархічних (вкладених) функцій, в першу чергу повинна бути визначена функція, що описує систему в цілому – контекстна функція. Функції зображуються на діаграмах як пойменовані прямокутники, або функціональні блоки (рис. 5.1).

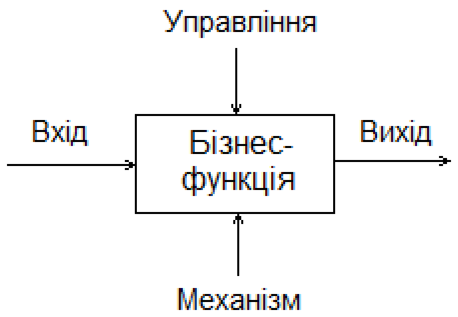


Рисунок 5.1 – Функціональний блок у IDEF0

Опис будь-якого функціонального блока має, як мінімум, включати в себе опис об’єктів, які блок створює в результаті своєї роботи («вихід»), та об’єктів, які блок споживає або перетворює («вхід»). Крім цього, в IDEF0 також моделюються управління та механізми виконання (рис. 5.1). Під управлінням розуміються об’єкти, що впливають на спосіб, яким блок перетворює вхід у вихід. Механізм виконання – це об’єкти, які безпосередньо виконують перетворення входу у вихід, але не споживаються при цьому самі по собі.

Для відображення категорій інформації, присутніх на діаграмах IDEF0, використовується абревіатура ICOM, що відображає чотири можливих типи стрілок:

- I (Input) – вхід – те, що споживається в ході виконання процесу;
- C (Control) – управління – обмеження та інструкції, що впливають на хід виконання процесу;
- O (Output) – вихід – результат виконання процесу;
- M (Mechanism) – виконуючий механізм – те, що застосовується для виконання процесу, але не споживається в цьому процесі.

Вихід функціонального блока може використовуватися в інших блоках. Основна цінність IDEF0 полягає в тому, що ця методологія допомагає виявити взаємозалежності між функціональними блоками системи. IDEF0 передбачає як розбиття, так і з’єднання стрілок на діаграмі.

Як правило, стрілки, що з'єднують функціональні блоки, мають найменування, що відрізняються від найменування вихідної стрілки. Вихідна і роз'єднані (або об'єднані) стрілки в сукупності називаються зв'язаними. Така техніка зазвичай застосовується для того, щоб відобразити використання в процесі тільки частини сировини або інформації, що позначаються вихідної стрілкою. Аналогічний підхід застосовується і до стрілок, що об'єднуються.

Поняття «пов'язані стрілки» використовується для управління рівнем деталізації діаграм. Якщо одна зі стрілок діаграми відсутня на батьківській діаграмі (наприклад, зважаючи на свою неістотність для батьківського рівня) і не пов'язана з іншими стрілками тієї ж діаграми, точка входу цієї стрілки на діаграму або виходу з неї позначається тунелем (табл. 4.1).

У IDEF0 реалізовані три базові принципи моделювання бізнес-процесів:

- принцип функціональної декомпозиції;
- принцип обмеження складності;
- принцип контексту.

Відповідно до принципу функціональної декомпозиції складна бізнес-функція може бути показана у вигляді сукупності складових її більш простих функцій, які самі у свою чергу можуть бути піддані декомпозиції.

Згідно з принципом обмеження складності кількість функціональних блоків на одній діаграмі повинна бути не менша двох і не більша шести. Таким чином забезпечується розбірливість і зручність використання діаграм IDEF0. Практика доводить, що дотримання цього принципу в більшості випадків приводить до того, що бізнес-процеси, подані у вигляді моделі IDEF0, добре структуровані, зрозумілі і легко піддаються аналізу.

Принцип контексту полягає в тому, що моделювання бізнес-процесу починається з побудови контекстної діаграми. На цій діаграмі відображається тільки один блок – головна бізнес-функція, що моделюється. При визначенні головної бізнес-функції необхідно завжди мати на увазі мету моделювання і точку зору на модель. Контекстна діаграма фіксує кордони модельованої бізнес-системи, визначаючи те, як моделююча система взаємодіє зі своїм оточенням. Це досягається за допомогою опису стрілок, з'єднаних з блоком, що являє головну бізнес-функцію.

Результатом застосування IDEF0 до деякої системи є модель цієї системи, що складається з ієрархічно впорядкованого набору діаграм, документації і словників, пов'язаних один з одним за допомогою перехресних посилань. До найбільш поширених CASE-засобів, що підтримує IDEF0 нотацію, відносять AllFusion Process Modeler (раніше BPwin), Microsoft Office Visio.

5.4.2. IDEF3-МОДЕЛЮВАННЯ ПОВЕДІНКИ СИСТЕМИ

IDEF3-моделювання органічно доповнює традиційне моделювання з використанням стандарту IDEF0. В даний час воно отримує все більше поширення як засіб побудови моделей бізнес-процесів.

Основою моделі IDEF3 служить так званий сценарій бізнес-процесу, який виділяє послідовність дій або підпроцесів аналізованої системи. Оскільки сценарієм обумовлені призначення і межі моделі, досить важливим є підбір відповідного найменування для позначення дій. Для підбору необхідного імені застосовуються стандартні рекомендації з кращим використання дієслів і дієслівних іменників.

Дія, або в термінах IDEF3 "одиниця роботи» (Unit of Work - UOW) – інший важливий компонент моделі. Діаграми IDEF3 відображають дію у вигляді прямокутника (рис. 5.2). Кожній з дій присвоюється унікальний ідентифікаційний номер. Цей номер не використовується знову навіть у тому випадку, якщо в процесі побудови моделі дія видаляється. У діаграмах IDEF3 номеру дії зазвичай передує номер її батьків.

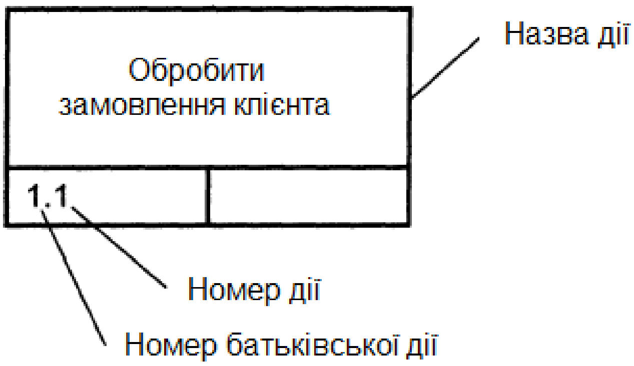


Рисунок 5.2 – Дія в IDEF3

Зв'язки виділяють істотні відношення між діями (табл. 5.2). Всі зв'язки в IDEF3 є односпрямованими, і, хоча стрілка може починатися або закінчуватися на будь-якій стороні блока, що позначає дію, діаграми IDEF3 зазвичай організовуються зліва направо таким чином, що стрілки починаються на правій і закінчуються на лівій стороні блоків.

Завершення однієї дії може ініціювати початок виконання відразу кількох інших дій, або, навпаки, певна дія може вимагати завершення кількох інших дій для початку свого виконання. З'єднання розбивають або з'єднують внутрішні потоки і використовуються для опису розгалуження процесу (табл. 5.3). Виділяють такі види з'єднань: розгортання і згортання. З'єднання розгортання використовується для розбиття потоку. Завершення однієї дії викликає початок виконання кількох інших. З’єднання згортання об’єднує потоки. Завершення одного або декількох дій викликає початок виконання тільки однієї іншої дії.

Таблиця 5.2 – Типи зв’язків у моделі IDEF3

Позначення	Назва	Опис
	Тимчасове передування (Temporal precedence)	Вихідна дія повинна завершитися перш, ніж кінцева дія зможе початися
	Об’єктний потік (Object flow)	Вихід вихідної дії є входом кінцевої дії. З цього, зокрема, випливає, що вихідна дія має завершитися перш, ніж кінцева дія зможе початися
	Нечітке відношення (Relationship)	Вид взаємодії між вихідною і кінцевою діями задається аналітиком окремо для кожного випадку використання такого відношення

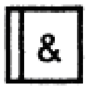
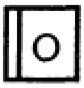
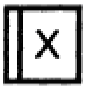
Таблиця 5.3 – Типи з’єднань у моделі IDEF3

Позначення	Назва	Вид	Опис
&	З’єднання «І»	Розгортаюче	Кожна кінцева дія обов’язково ініціюється
		Згортаюче	Кожна вихідна дія обов’язково має завершитися
X	З’єднання «Ексклю-зивне АБО»	Розгортаюче	Одна і тільки одна кінцева дія ініціюється
		Згортаюче	Одна і тільки одна вихідна дія повинна завершитися
O	З’єднання «АБО»	Розгортаюче	Одна (або більше) кінцева дія ініціюється
		Згортаюче	Одна (або більше) вихідна дія повинна завершитися

Є випадки, коли час початку або закінчення паралельно виконуваних дій має бути однаковим, тобто дії повинні виконуватися синхронно. Для моделювання такого поводження системи використовуються синхронні з’єднання (табл. 5.4).

Для залучення уваги користувача до будь-яких важливих аспектів моделі на діаграму виносяться покажчики. Покажчики – це спеціальні символи, які посилаються на інші розділи опису процесу (табл. 5.5).

Таблиця 5.4 – Синхронні з’єднання в моделі IDEF3

Позначення	Назва	Вид	Опис
	І	Розгортаюче	Усі дії почнуться одночасно
		Згортаюче	Всі дії закінчаться одночасно
	АБО	Розгортаюче	Можливо, кілька дій почнуться одночасно
		Згортаюче	Можливо, кілька процесів закінчаться одночасно
	Ексклю-зивне АБО	Розгортаюче	Одночасний початок дій неможливий
		Згортаюче	Одночасне закінчення дій неможливе

Таблиця 5.5 – Типи покажчиків в моделі IDEF3

Тип покажчика	Опис
ОБ’ЄКТ (OBJECT)	Для опису того, що в дії бере участь об’єкт, який заслуговує окремої уваги
ПОСИЛАННЯ (GOTO)	Для реалізації циклічності виконання дій. Даний вказівник може відноситися і до з’єднання
ОДИНИЦЯ ДІЇ (Unit of Behavior)	Для розміщення на діаграмі додаткового примірника вже існуючої дії без зациклення
ЗАМІТКА (NOTE)	Для документування будь важливої інформації загального характеру, що стосується зображеного на діаграмах. У цьому сенсі ПОСИЛАННЯ служить альтернативою методу розміщення текстових нотаток безпосередньо на діаграмах
УТОЧНЕННЯ (Elaboration – ELAB)	Для уточнення або більш докладного опису зображеного на діаграмі. Зазвичай використовуються для опису логіки розгалуження у з’єднаннях

Дії в IDEF3 можуть бути декомпоновані, або розкладені на складові, для більш детального аналізу. Декомпонувати дію можна кілька разів. Це дозволяє документувати альтернативні потоки процесу в одній моделі.

Для створення діаграм IDEF3 можуть бути використані такі CASE-засоби як AllFusion Process Modeler, Microsoft Office Visio.

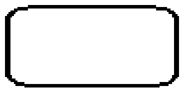
5.4.3. МОДЕЛЮВАННЯ ПОТОКІВ ДАНИХ

Так само, як і діаграми IDEF0, діаграми потоків даних DFD моделюють систему як набір дій, з’єднаних одна з одною стрілками. Діаграми потоків даних також можуть містити два нових типи об’єктів: об’єкти, що збирають і зберігають інформацію – «сховища даних», і «зовнішні сутності» – об’єкти, які моделюють взаємодію з частинами системи (або іншими системами), що виходять за межі моделювання (табл. 5.6).

На відміну від стрілок у IDEF0, які ілюструють відносини, стрілки в DFD показують, як об’єкти (включаючи дані) реально переміщуються від однієї дії до іншої. Це подання разом зі сховищами даних і зовнішніми сутностями забезпечує відображення в DFD-моделях таких фізичних характеристик системи, як рух об’єктів (потоки даних), зберігання об’єктів (сховища даних), джерела і споживачі об’єктів (зовнішні сутності).

На відміну від IDEF0, що розглядає систему як множину взаємопересічних дій, у назвах об’єктів DFD-діаграм переважають іменники. Контекстна DFD-діаграма часто складається з одного функціонального блока і декількох зовнішніх сутностей. Функціональний блок на цій діаграмі зазвичай має ім’я, що збігається з ім’ям усієї системи.

Таблиця 5.6 – Елементи DFD-діаграми

Позначення	Назва	Опис
	Функція (Function, Process)	Функціональний блок перетворює вхідні дані у вихідні
	Сховища даних (Datastore)	Являють будь-який механізм, який підтримує зберігання даних для їх проміжної обробки
	Зовнішні сутності (External Entities)	Зовнішні сутності забезпечують необхідні входи для системи та/або є приймачами для її виходів. Одна зовнішня сутність може одночасно надавати входи (функціонуючи як постачальник) і приймати виходи (функціонуючи як одержувач)
	Потік даних (Dataflow)	Стрілки описують пересування (потік) об’єктів від однієї частини системи до іншої. Також використовуються двонаправлені стрілки, які потрібні для відображення взаємодії між блоками

Стрілки на DFD-діаграмах можуть бути розбиті (розгалужені) на частини, і при цьому кожен вихідний сегмент може бути перейменований таким чином, щоб показати декомпозицію даних, які переносяться потоком. Також стрілки можуть з’єднуватися між собою для формування так званих комплексних об’єктів.

До CASE-засобів, що підтримують нотацію DFD, відносять AllFusion Process Modeler і Microsoft Office Visio.

Таким чином, діаграми IDEF0, IDEF3 і DFD є зручним інструментом для моделювання бізнес-процесів. Кожна з цих нотацій може використовуватися як самостійно, так і спільно з іншою, наприклад, будь-який функціональний блок IDEF0 може бути поданий у вигляді послідовності процесів засобами IDEF3.

5.5. Об'єктно-орієнтований аналіз та проектування бізнес-процесів

UML – це наступник того покоління методів об'єктно-орієнтованого аналізу і проектування (ООАП), які з'явилися наприкінці 1980-х і початку 1990-х років. Головними в розробленні UML були наступні цілі:

- надати користувачам готову до використання виразну мову візуального моделювання, що дозволяє їм розробляти осмислені моделі й обмінюватися ними;
- передбачити механізми розширюваності та спеціалізації для розширення базових концепцій;
- забезпечити незалежність від конкретних мов програмування і процесів розробки;
- забезпечити формальну основу для розуміння цієї мови моделювання (мова повинна бути одночасно точною і доступною для розуміння, без зайвого формалізму);
- стимулювати зростання ринку об'єктно-орієнтованих інструментальних засобів;
- інтегрувати кращий практичний досвід.

UML прийнятий в якості стандартної мови моделювання, що отримала широку підтримку в індустрії програмного забезпечення.

Головна перевага об'єктно-орієнтованого підходу (ООП) формулюється таким чином: об'єктно-орієнтовані системи більш відкриті і легше піддаються внесенню змін, оскільки їх конструкція ґрунтується на стійких формах. Це дає можливість системі розвиватися поступово і не призводить до повної її переробки навіть у разі істотних змін. Крім цього, використання ООП суттєво збільшує рівень уніфікації розробки і придатність для повторного використання.

Об'єктна модель цілком природна, оскільки в першу чергу орієнтована на людське сприйняття світу. Також об'єктна модель дозволяє повною мірою використовувати виразні можливості об'єктно-орієнтованих мов програмування.

До недоліків ООП відносять високі початкові витрати. Об'єктна декомпозиція істотно відрізняється від функціональної, тому перехід на нову технологію пов'язаний як з подоланням психологічних труднощів, так і з додатковими фінансовими витратами.

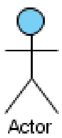
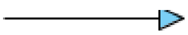
Таким чином, структурний підхід, як і раніше, зберігає свою значущість і досить широко використовується на практиці. На прикладі мови UML добре видно, що її автори запозичили те раціональне, що можна було взяти із структурного підходу: елементи функціональної декомпозиції в діаграмах варіантів використання, діаграми станів, діаграми діяльності та ін. Зрозуміло, що в конкретному проекті складнішої системи неможливо обійтися тільки одним способом декомпозиції.

Для моделювання бізнес-процесів найчастіше використовуються діаграми прецедентів і діаграми діяльності (табл. 5.7, 5.8).

Діаграми прецедентів (Use-case diagram) відіграють основну роль у моделюванні поведінки системи, підсистеми або класу. Даний вид діаграми показує множину прецедентів, акторів і відношень між ними (табл. 5.7).

Між прецедентами можливі відношення включення (Include) і розширення (Extend). Відношення включення означає, що в деякій точці базового прецеденту інкорпоровано поведінку іншого прецеденту. Прецедент, що включається, ніколи не існує автономно. Відношення розширення має на увазі, що базовий прецедент неявно містить поведінку іншого прецеденту в точці, яка неявно задається прецедентом, що розширює. Базовий прецедент може бути автономним, але за певних обставин його поведінка розширюється за рахунок іншого.

Таблиця 5.7 – Елементи діаграми прецедентів

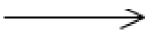
Позначення	Назва	Опис
	Прецедент (Use-case)	Опис множини послідовностей дій, виконуваних для того, щоб актор міг отримати певний результат
	Актор (Actor)	Зв’язана множина ролей, які користувачі прецедентів виконують під час взаємодії з ними
	Відношення залежності (Dependency)	Відношення, згідно з яким зміна одного елементу може вплинути на інший елемент, який його використовує, причому зворотне не є обов’язковим
	Відношення узагальнення (Generalization)	Відношення між узагальненою сутністю (батьками) і її конкретним втіленням (нащадком)
	Відношення асоціації (Association)	Структурне відношення, яке показує, що об’єкти одного типу пов’язані з об’єктами іншого типу

Для моделювання бізнес-процесів і робочих потоків можуть використовуватися діаграми діяльності (Activity diagram). Діаграма діяльності складається зі станів діяльності і станів дії, переходів і об’єктів (табл. 5.8).

Стани дії не можуть бути піддані декомпозиції, вони атомарні. Це означає, що всередині них можуть відбуватися події, але виконувана в дії робота не може бути перервана.

На противагу цьому діяльність може бути піддана подальшій декомпозиції, внаслідок чого виконувану діяльність можна подати за допомогою інших діаграм діяльності. Стан діяльності не є атомарними, тобто він може бути перерваний.

Таблиця 5.8 – Елементи діаграми діяльності

Позначення	Назва	Опис
	Діяльність (Activity)	Неатомарний крок процесу, що триває у часі
	Дія (Action)	Атомарний крок, що змінює стан системи або повертає якесь значення
	Розгалуження (Decision node)	Описує різні шляхи виконання залежно від значення деякого булевого виразу
	Початковий вузол (Initial node)	Позначає початок процесу
	Кінцевий вузол (Activity final node)	Позначає завершення процесу
	Потік виконання / Потік об’єктів (Control flow / Object flow)	Задає напрям управління або руху об’єктів (даних)

При моделюванні бізнес-процесів у діаграмі діяльності можливе використання паралельних потоків. Для позначення поділу і злиття паралельних потоків виконання використовується синхронізаційна риска (жирна вертикальна або горизонтальна лінія).

У діаграмі діяльності можливе використання «доріжок» (swim lanes). Вони дозволяють розбити діяльності на групи, кожна з яких є відділом, що відповідає за певну роботу.

Таким чином, за допомогою засобів UML можливе моделювання бізнес-процесів. Основні переваги вибору даної технології проявляються при автоматизації бізнес-процесів. Вибір інструментальних засобів, що підтримують UML-моделювання, досить широкий, найбільш поширені Rational Rose, Visual Paradigm, Sybase Power Designer, ER/Studio Business Architect.

5.6. Графічне подання бізнес-процесів на основі BPMN

BPMN (Business Process Model and Notation) дає змогу користувачам розробляти легко зрозумілі графічні зображення різних бізнес-процесів. BPMN надає безліч діаграм, спеціально розроблених для людей, які займаються проектуванням і управлінням бізнес-процесів.

На сьогоднішній день різні підприємства та організації використовують безліч засобів зображення бізнес-процесів. Тому бізнес-аналітику необхідно розуміти і розбиратися в різних графічних нотаціях моделювання бізнес-процесів протягом їхнього життєвого циклу, що включає їх створення, виконання, моніторинг й аналіз. Тому стандартна графічна нотація BPMN значно допомагає в розумінні взаємодій і транзакцій як усередині одного підприємства, так і між організаціями. BPMN надає можливість підприємствам моделювати їх внутрішні бізнес-процеси у графічній нотації. Це забезпечує розуміння між учасниками бізнес-процесів і дозволяє швидко пристосовуватись до нових обставин. BPMN успадковує традиціїї позначень блок-схем для легкого сприйняття і гнучкого редагування.

За допомогою діаграм BPMN можна моделювати процеси з точки зору різних учасників. Деякі дії є внутрішніми відносно одного учасника, оскільки він їх виконує сам або контролює їх виконання. Інші дії є зовнішніми щодо цього учасника. Хоча точка зору, з якої розглядається бізнес-процес, важлива, BPMN поки не пропонує графічних механізмів для її визначення.

Існує п'ять категорій елементів у нотації BPMN:

1. Основні об'єкти або елементи потоку (Flow objects):
- події (Events);
- дії (Activities);
- шлюзи (Gateways).

2. Дані (Data):
- об'єкти даних (Data Objects);
- дані входів (Data Inputs);
- дані виходів (Data Outputs);
- сховища даних (Data Stores).

3. З'єднувальні елементи (Connecting Objects):
- послідовності (Sequence Flows);
- повідомлення (Message Flows);
- асоціації (Associations);
- асоціації даних (Data Associations).

4. Структуруючі елементи (Swimlanes):

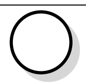
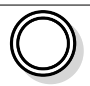
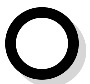
- області (Pools);
- доріжки (Lanes).

5. Артефакти (Artifacts):

- групи (Groups);
- нотатки (Text Annotations).

У BPMN існує три типи подій: початкова, проміжна, кінцева (табл. 5.9). Події впливають на потік процесу, вони можуть бути причиною або результатом процесу.

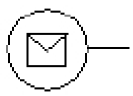
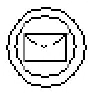
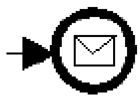
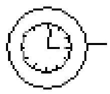
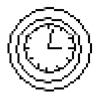
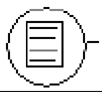
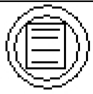
Таблиця 5.9 – Типи подій в моделі BPMN

Позначення	Назва	Опис
	Початкова подія (Start Event)	Починає потік процесу
	Проміжна подія (Intermediate Event)	Трапляється в ході потоку процесу
	Кінцева подія (End Event)	Завершує потік процесу

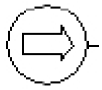
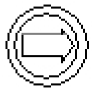
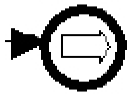
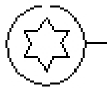
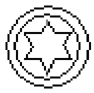
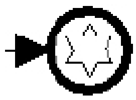
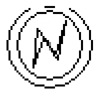
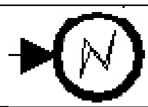
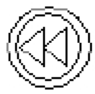
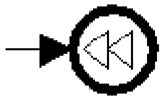
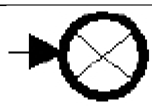
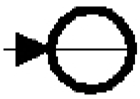
Для моделювання більш складних бізнес-подій, таких як повідомлення, таймери, бізнес-правила, умови помилок, використовуються тригери (табл. 5.10).

У BPMN існує три види дій (Activities) – сам процес (process), підпроцес (sub-process) (рис. 5.3), завдання (task). Їх назви відображають ієрархічність відносин між ними. Кожен з них зображається за допомогою прямокутника. Завдання не піддаються декомпозиції.

Таблиця 5.10 – Типи тригерів у моделі BPMN

Тригер	Початкова подія	Проміжна подія	Кінцева подія	Опис
Повідом-лення (Message)				Повідомлення може розпочинати процес, продовжувати або завершувати його
Таймер (Timer)			Не може бути	Певний час або цикл можуть починати або продовжувати процес
Правило (Rule)			Не може бути	Спрацьовує, коли умова в правилі виконується

Закінчення таблиці 5.10

Тригер	Початкова подія	Проміжна подія	Кінцева подія	Опис
З'єднання (Link)				Механізм, що зв'язує кінцеву подію одного потоку процесу з початковою подією іншого потоку процесу
Множинна подія (Multiple)				Є кілька тригерів для потоку процесу
Виняток (Exception)	Не може бути			Названий виняток має бути згенерований
Компенса-ція (Compen-sation)	Не може бути			Процес вимагає компенсації
Закінчення (End)	Не може бути	Не може бути		Користувач вирішив закінчити процес
Закриття (Kill)	Не може бути	Не може бути		Відбулася фатальна помилка і всі дії процесу повинні бути негайно завершені

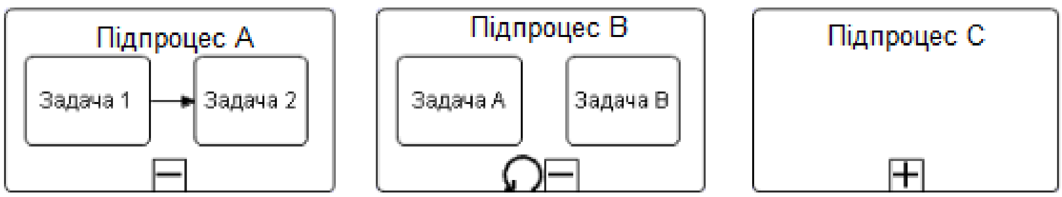
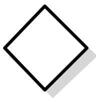
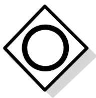
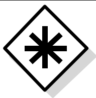
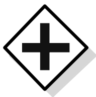


Рисунок 5.3 – Приклади підпроцесів (A – звичайний розгорнутий, B – повторюваний розгорнутий, C – звичайний прихований)

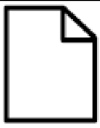
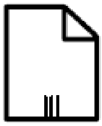
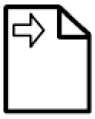
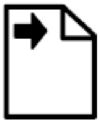
Для позначення точок прийняття рішень на діаграмі використовуються шлюзи. Шлюзи можуть визначати напрямок потоку на основі даних (data-based) або на основі настання події (event-based). У табл. 5.11 наведені основні види шлюзів.

Об'єкти даних (Data Objects) надають інформацію про те, що є необхідним для дії задля її виконання, або про те, що виробляється процесом. Об'єкт даних може показувати одиничний об'єкт або набори об'єктів. Дані входу і виходу надають відповідну інформацію для процесу. Об'єкти даних наведені в табл. 5.12.

Таблиця 5.11 – Види шлюзів у моделі BPMN

Позначення	Назва	Опис
	Виключаюча умова Exclusive Decision (XOR)	Потік процесу йде тільки в одному напрямку, який задовольняє умові
	Включаюча умова Inclusive OR Decision	Можливий один або більше вихідних потоків
	Складна умова Complex Decision	Задається складна умова, що посилається на вихідні потоки
	Паралельне розгалуження Parallel Forking (Joining)	Всі вихідні (вхідні) потоки задіяні

Таблиця 5.12 – Види об’єктів даних у моделі BPMN

Позначення	Назва	Опис
	Одиничний об’єкт даних (Singular Data Object)	Дані, що використовуються в процесі, подані одним об’єктом
	Об’єкт даних, що показує колекцію об’єктів (Data Object that is a collection)	Дані, що використовуються в процесі, подані кількома об’єктами
	Сховище даних (Data Store)	Надає механізм для вилучення та оновлення інформації, яка буде зберігатися поза бізнес-процесом
	Дані входу (Data Input)	Подає дані на вході процесу
	Дані виходу (Data Output)	Подає дані на виході процесу

З’єднувальні елементи, або конектори (Connecting Objects) використовуються для з’єднання елементів діаграми. Види конекторів наведені в табл. 5.13.

Таблиця 5.13 – Види з’єднувальних елементів у моделі BPMN

Позначення	Назва	Опис
⟶	Послідовний потік Sequence Flow	Вказує напрямок потоку процесу
····>	Потік повідомлень Message Flow	Відображає потік повідомлень між елементами
......	Асоціація Association	Вказує на асоціацію між елементами

Області і доріжки (Swimlanes) групують процес за категоріями виконавців та учасників процесу. Артефакти подають текстові коментарі (Text Annotations) і групують дії, не відбиваючись на потоці процесу (Group).

До CASE-засобів, що підтримують моделювання бізнес-процесів в нотації BPMN, відносять Visual Paradigm, BPMN 2.0 Modeler для Microsoft Office Visio.

5.7. Огляд Visual Paradigm for UML

Visual Paradigm for UML (VP-UML) – це потужний, міжплатформний, легкий у використанні засіб візуального моделювання UML діаграм. VP-UML забезпечує розробників зручною платформою для побудови якісних програмних рішень у швидший, кращий та дешевший спосіб. VP-UML дозволяє взаємодіяти з іншими CASE засобами та інтегрованими середовищами розробки, необхідними для процесів моделювання, кодування та розгортання програмного забезпечення.

На рис. 5.4 зображено, як виглядає середовище Visual Paradigm for UML Enterprise Edition. У табл. 5.14 наведені елементи вікна середовища VP-UML та подано їх опис.

Таблиця 5.14 – Елементи середовища VP-UML

№ п/п	Елемент	Опис
1	Menu bar	Полоска меню вгорі вікна дозволяє обирати та здійснювати різні операції у Visual Paradigm for UML
2	Toolbar	Панель інструментів, що знаходиться нижче меню, є його розширенням. Усі кнопки зібрані у групи для зручності використання
3	Diagram Navigator	Поле, де розміщено список діаграм та де можна створити та знайти діаграми залежно від їхнього типу

Закінчення таблиці 5.14

№ п/п	Елемент	Опис
4	Properties Pane	Поле, де відображаються властивості обраної моделі
5	Message Pane	Місце, де вказується інформація або можливі попередження
6	Diagram Pane	Діаграма відображається у полі діаграм

VP-UML призначений для широкого кола користувачів, включаючи розробників програмного забезпечення, системних аналітиків, бізнес-аналітиків, проектувальників системної архітектури та усіх, хто зацікавлений у надійному розробленні програмних рішень, що базуються на об’єктно-орієнтованому підході.

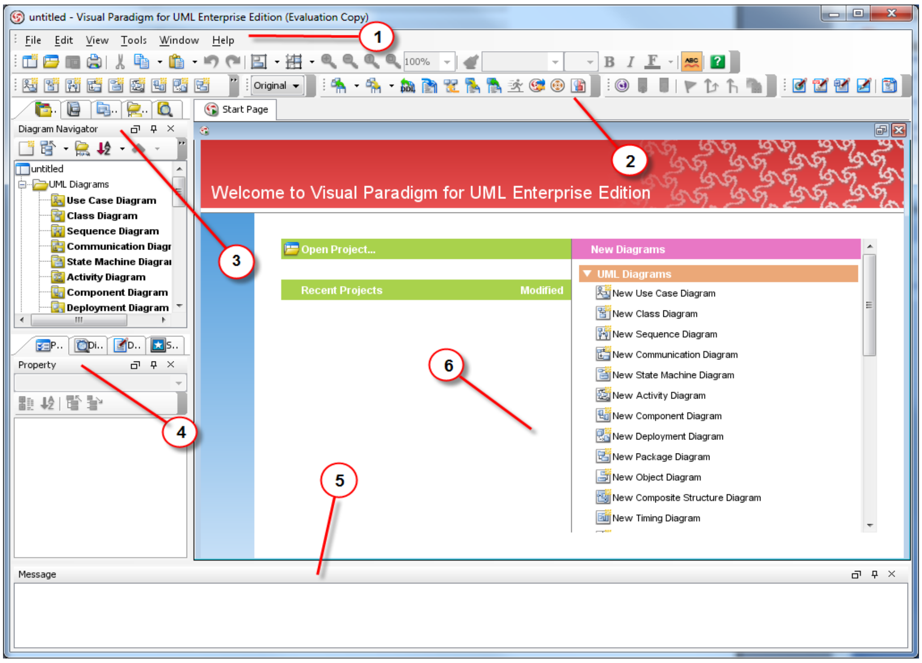


Рисунок 5.4 – Visual Paradigm for UML Enterprise Edition

Щоб створити діаграму, наприклад, діаграму прецедентів, необхідно:

1) Натиснути правою кнопкою на Use Case Diagram у Diagram Navigator та вибрати New Use Case Diagram (рис. 5.5).

2) Ввести назву діаграми.

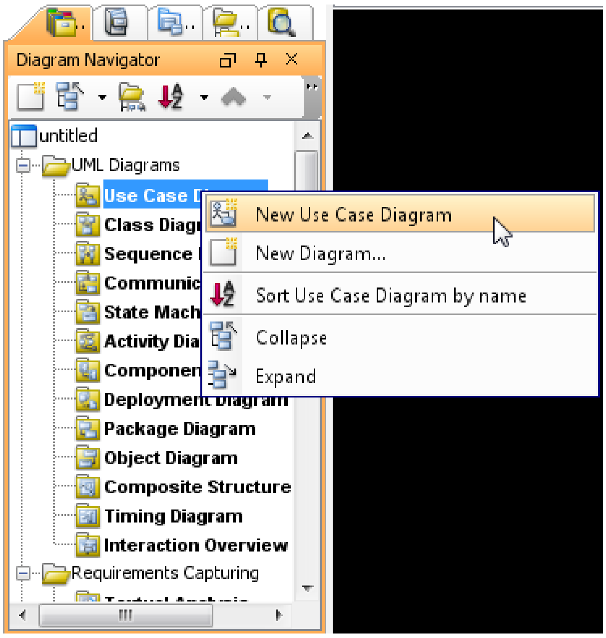


Рисунок 5.5 – Створення нової діаграми

Щоб створити певний елемент, треба вибрати тип елемента в інструментах діаграм, натиснути на діаграмі та визначити необхідний розмір. Наприклад, для створення прецеденту необхідно:

1) Вибрати Use Case з інструментів діаграми, натиснути на діаграмі та визначити розмір (рис. 5.6).

2) Ввести назву прецеденту та натиснути Enter для її підтвердження (рис. 5.7)

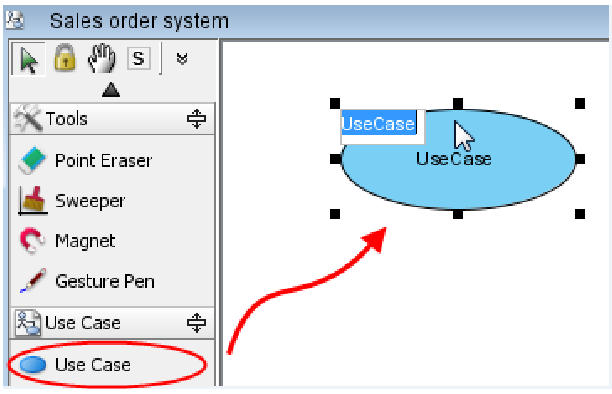


Рисунок 5.6 – Створення прецеденту

Два елементи можуть бути пов’язані за допомогою значків, що оточують кожний елемент. Наприклад, якщо необхідно зробити асоціацію між актором та прецедентом, слід навести мишку на актора, натиснути на значок Association, навести на прецедент та відпустити кнопку миші (рис. 5.8).

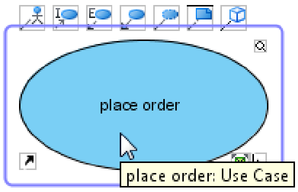


Рисунок 5.7 – Введення назви прецеденту

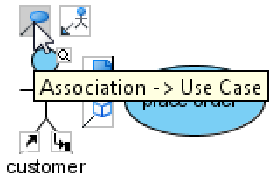


Рисунок 5.8 – Асоціація між актором та прецедентом

Для документування будь-якого елемента можна ввести його текстовий опис, відкривши Documentation Pane у нижній лівій частині вікна (рис. 5.9).

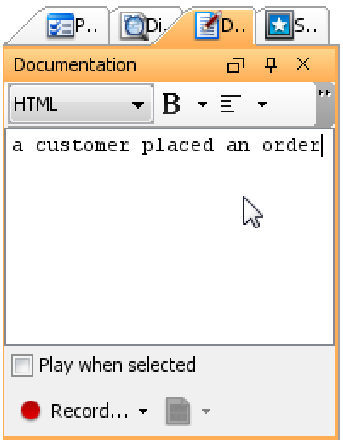


Рисунок 5.9 – Текстовий опис

Відповідно до своїх уподобань можна форматувати вигляд елементів, натиснувши правою кнопкою миші на елемент та вибравши Styles and Formatting, потім – Formats…. У такий саме спосіб можна форматувати зв'язки між елементами.

Можна змінити фон прецедентів, наприклад:

1) Натиснути правою кнопкою миші на прецедент та вибрати Styles and Formatting, потім – Formats… (рис. 5.10).

2) У діалоговому вікні Formats вибрати колір, наприклад, зелений, та натиснути «ОК» для підтвердження зміни (рис 5.11).

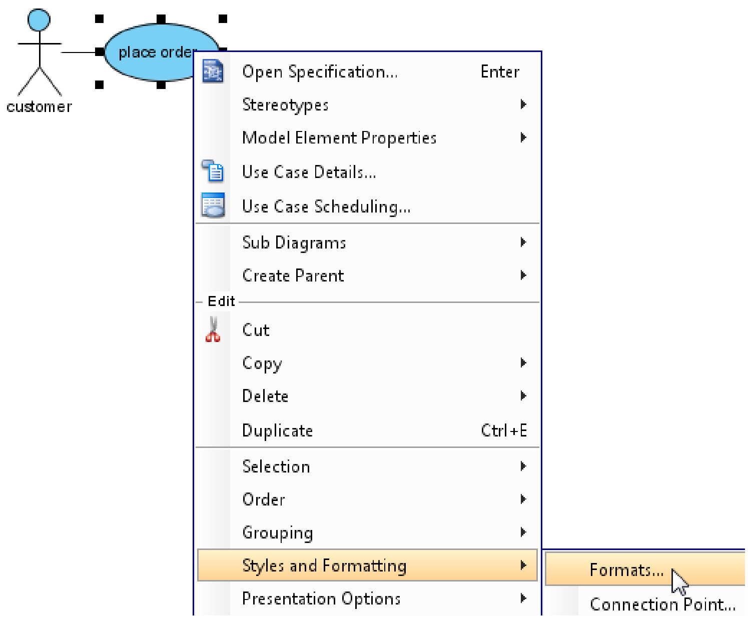


Рисунок 5.10 – Форматування прецеденту

Рівень – це така властивість, що дозволяє розділяти елементи діаграми на логічні групи та виконувати різні дії, включаючи зміну їхньої видимості або заборону редагування.

Щоб створити новий рівень, треба вибрати View > Layers… у головному меню. У діалоговому вікні Diagram Layers натиснути на кнопку Create new layer. Назвати новий рівень Annotation Layer та натиснути Close для підтвердження зміни (рис. 5.12).

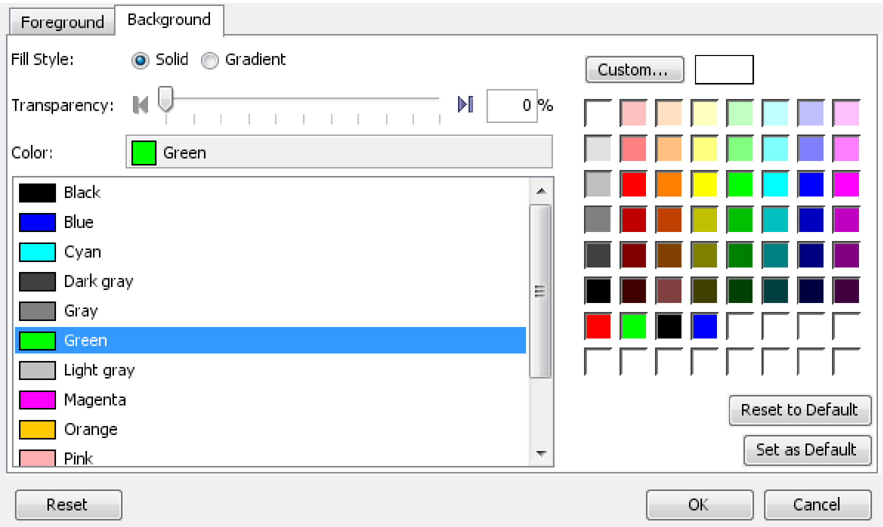


Рисунок 5.11 – Зміна кольору прецеденту

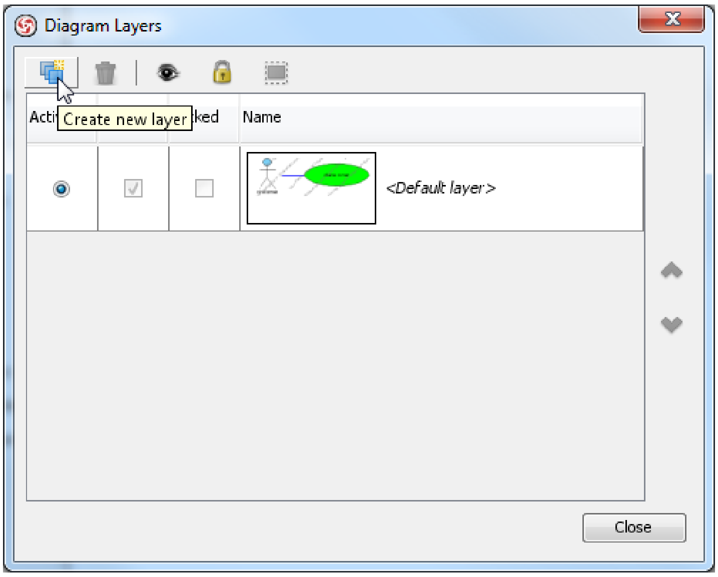


Рисунок 5.12 – Створення нового рівня

Елемент може бути переведений на інший рівень за допомогою правого натискання на нього та вибору рівня, на який треба його перевести.

Для побудови діаграми прецедентів необхідно визначити акторів та самі прецеденти. Отже, спочатку задаємо акторів (рис. 5.13).

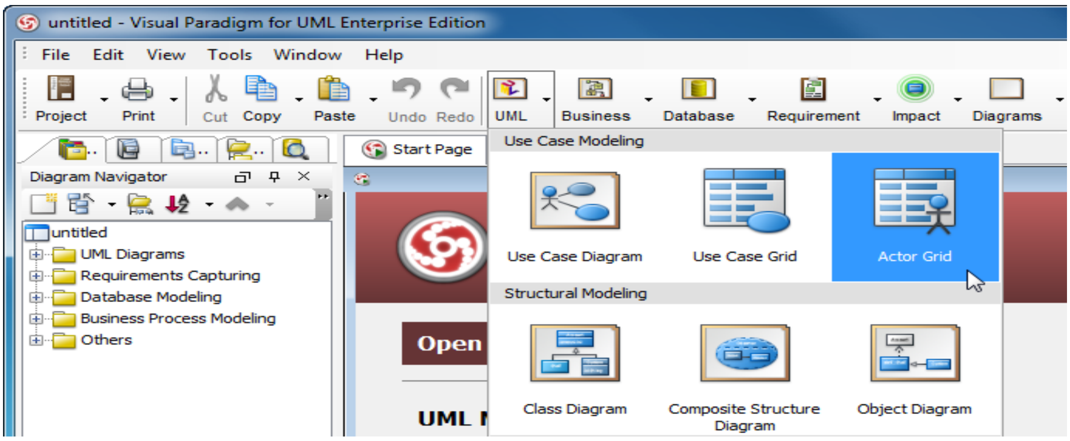


Рисунок 5.13 – Створення акторів

Необхідно виконати наступну послідовність дій.

1) Натиснути кнопку Create Actor у верхньому лівому куті.

2) Ввести ім’я актора (рис. 5.14).

3) Задокументувати актора у полі Documentation (рис. 5.15).

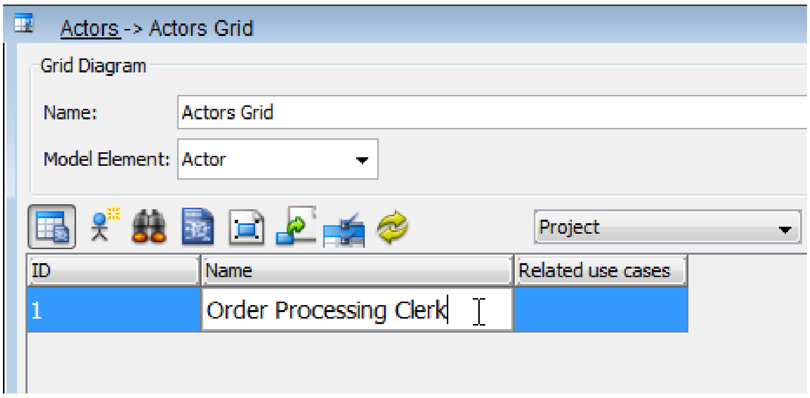


Рисунок 5.14 – Введення ім'я актора

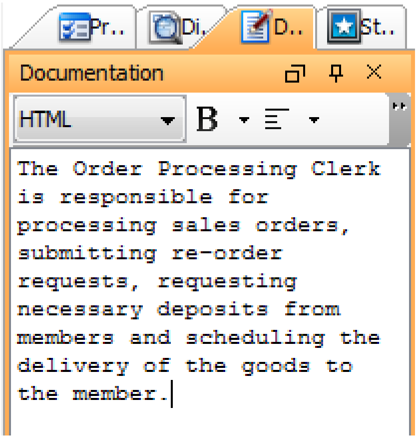


Рисунок 5.15 – Документування актора

Після створення акторів можна створити прецеденти, обравши UML > Use Case Grid (рис. 5.16).

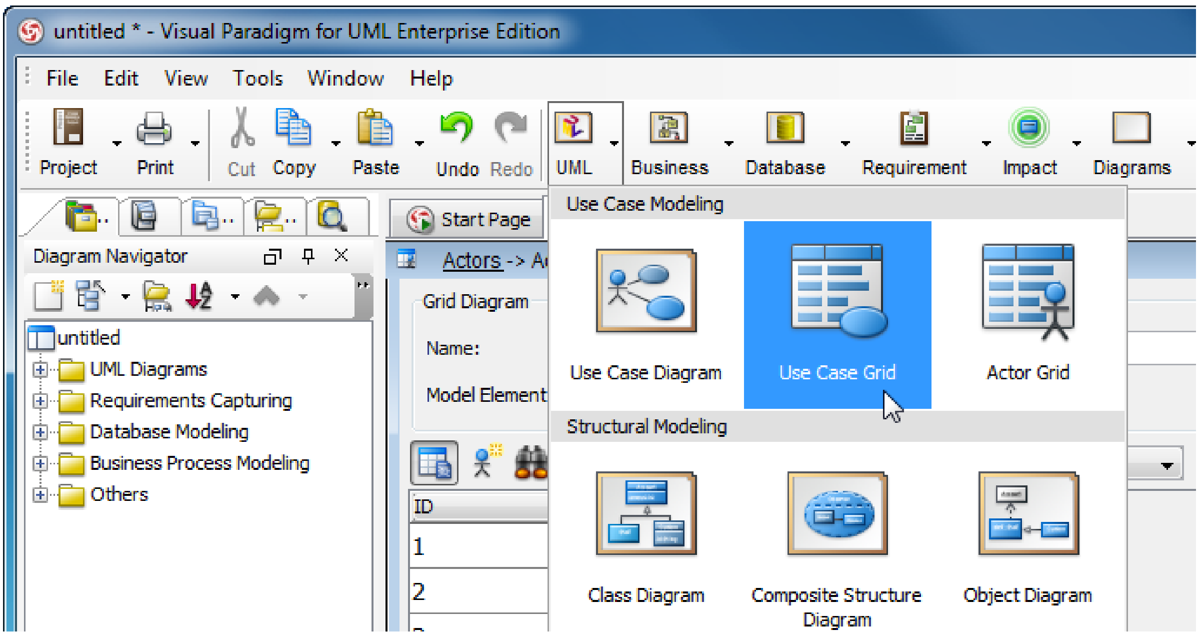


Рисунок 5.16 – Створення прецедентів

Далі треба виконати наступні дії:

1) За допомогою кнопки Create Use Case у верхньому лівому куті створити прецедент.

2) Ввести назву прецеденту (рис. 5.17).

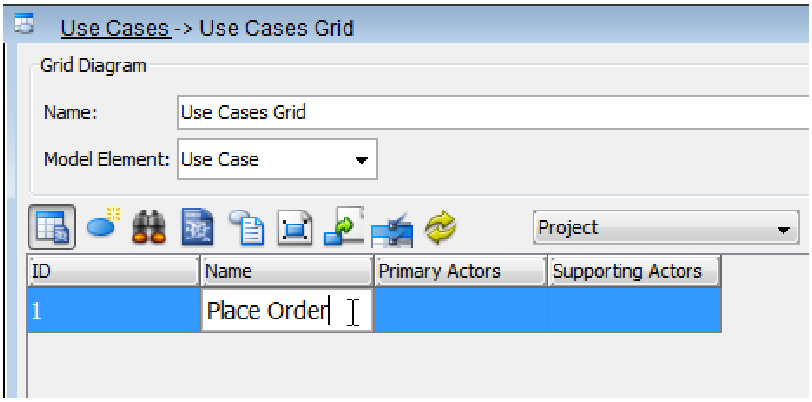


Рисунок 5.17 – Введення назви прецеденту

Для побудови діаграми прецедентів треба виконати наступне:

1) Вибрати прецедент з Use Case Grid та натиснути кнопку Visualize (рис. 5.18).

2) У діалоговому вікні Visualize Model Element вибрати опцію Create new diagram (рис. 5.19).

3) Відкрити Model Explorer та пересунути створеного актора у поле діаграми (рис. 5.20).

4) Оскільки даний актор не з'являвся на жодній діаграмі досі, необхідно задати його як Master view у проекті (рис. 5.21).

5) З'єднати актора та прецедент зв'язком асоціації (рис. 5.22).

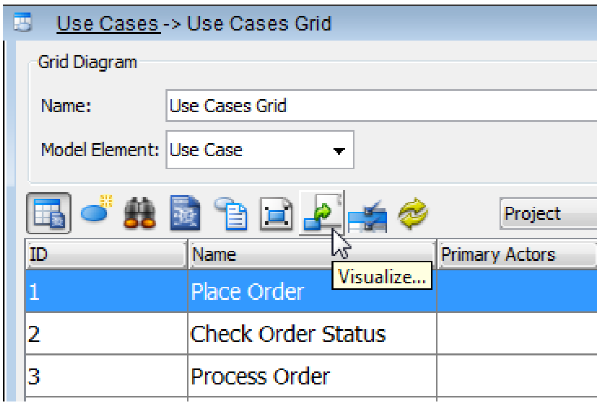


Рисунок 5.18 – Візуалізація прецеденту

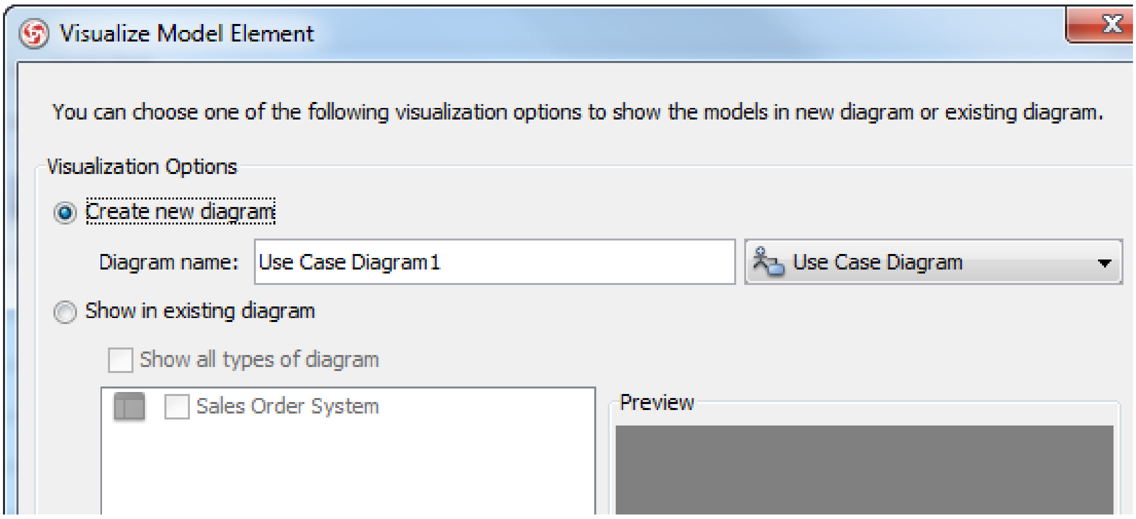


Рисунок 5.19 – Створення нової діаграми прецедентів

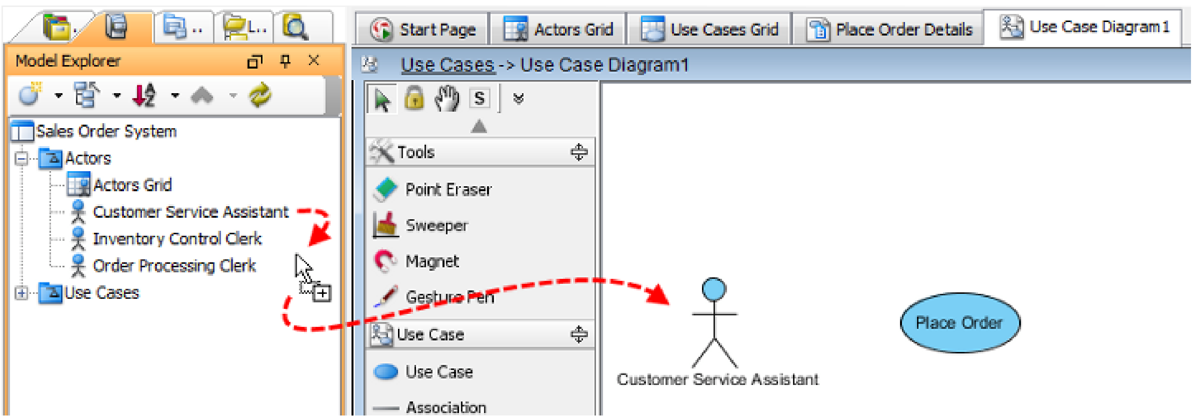


Рисунок 5.20 – Візуалізація актора

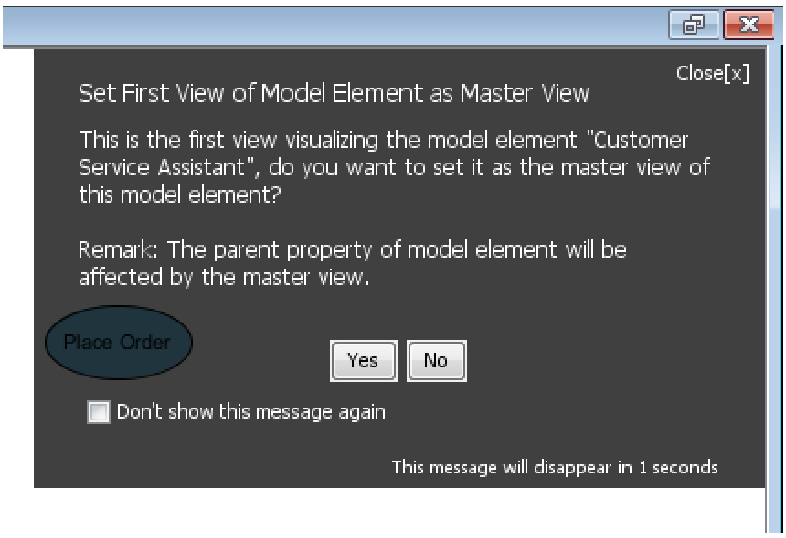


Рисунок 5.21 – Специфікація актора

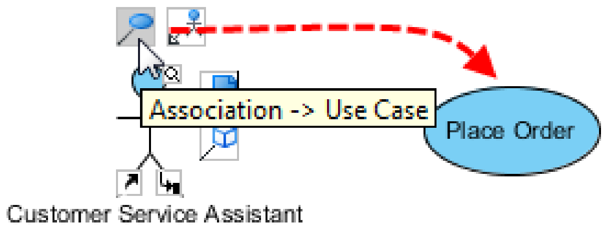


Рисунок 5.22 – Зв'язок між актором та прецедентом

Глосарій – це місце, де зберігаються поняття, що є специфічними для заданої предметної області. Замість додавання понять з уяви можна брати поняття з потоку подій. Наприклад:

1) Припустимо, «online system homepage» є ключовою фразою певного потоку подій. Необхідно натиснути правою кнопкою миші на цьому словосполученні та вибрати Add «online system homepage» to Glossary з меню (рис. 5.23).

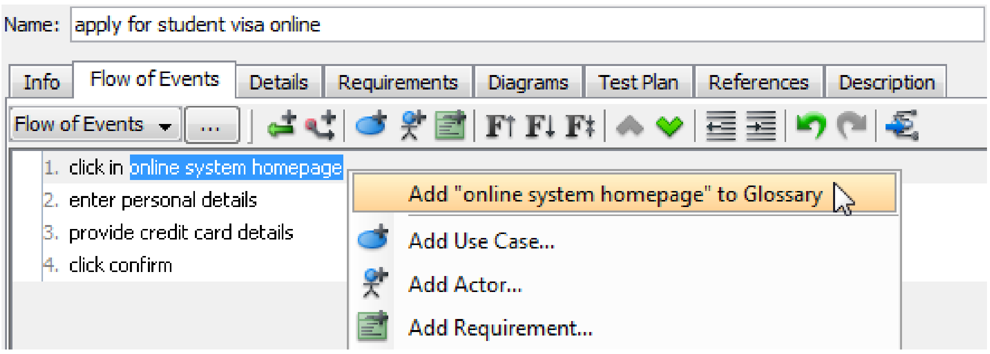


Рисунок 5.23 – Додавання поняття до глосарію

2) Відкриється глосарій. Натиснувши правою кнопкою миші на понятті, треба вибрати його специфікацію – Open online system homepage Specification (рис. 5.24).

3) Необхідно визначити поняття, близькі до даного. Потім додати опис та натиснути «ОК» для завершення редагування (рис. 5.25).

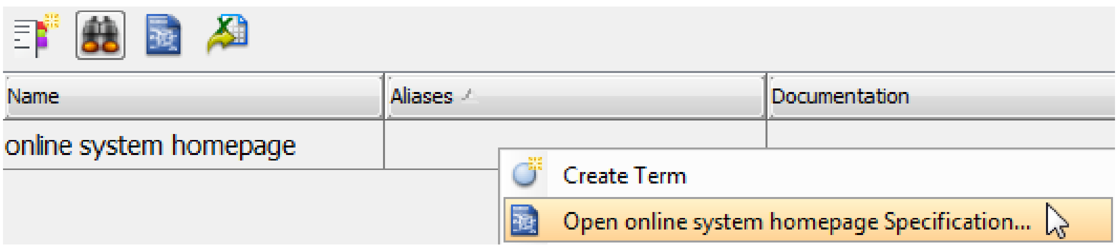


Рисунок 5.24 – Відкриття специфікації поняття

Діаграма послідовності відображає взаємозв’язки між об’єктами, показані як лінії життя у послідовному порядку. Ці лінії показують усі точки взаємодії з іншими об’єктами подій. Діаграма послідовності може бути створена натисканням правої кнопки миші на Sequence Diagram у Diagram Navigator та вибором New Sequence Diagram.

Наприклад, для прецеденту «apply for student visa online» слід зробити наступні кроки:

1) Вибрати актора з панелі інструментів.

2) Натиснути значок Message -> LifeLine та перетягти його у бажане місце для створення лінії життя, пов’язаної з актором. Назвати лінію життя та повідомлення (рис. 5.26).

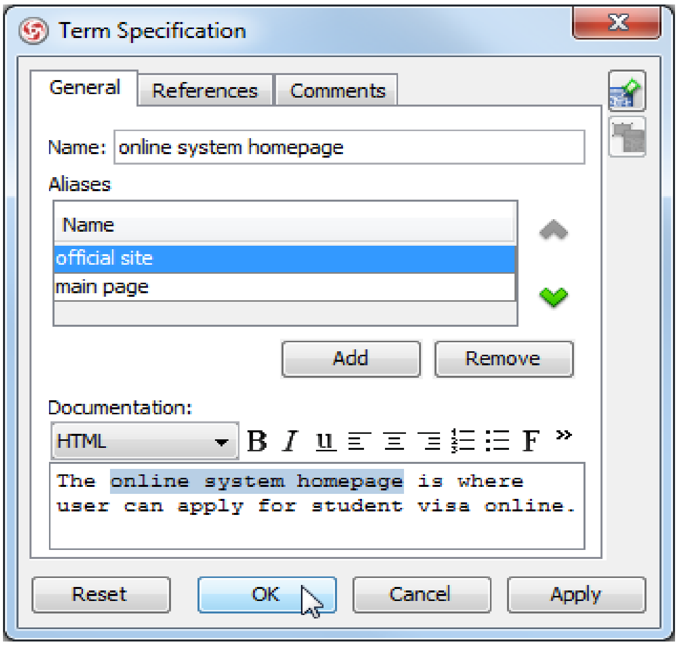


Рисунок 5.25 – Редагування поняття

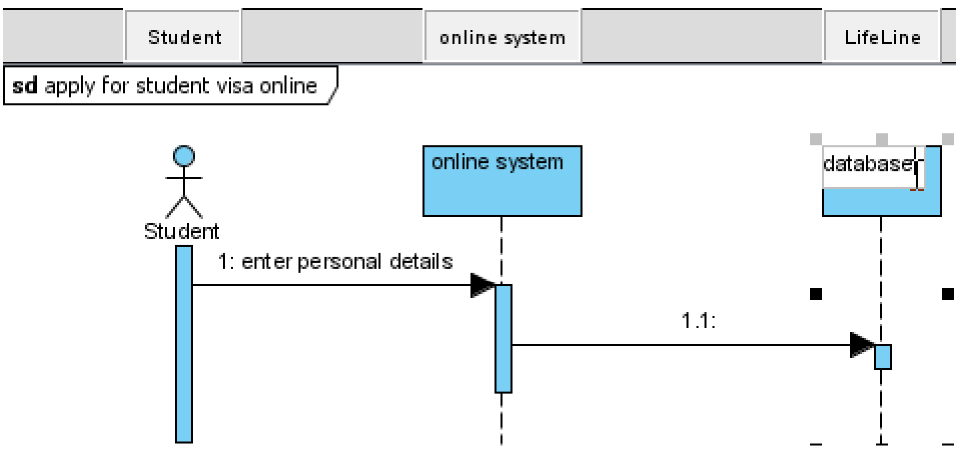


Рисунок 5.26 – Створення діаграми послідовності

Діаграма діяльності показує контрольний потік від однієї дії до іншої. Вона може моделювати динамічні аспекти системи. Щоб створити нову діаграму діяльності, треба натиснути на Activity Diagram правою кнопкою миші у Diagram Navigator та вибрати New Activity Diagram.

Далі треба виконати наступну послідовність дій:

1) Вибрати Initial Node на панелі інструментів та перенести її в область діаграм (рис. 5.27).

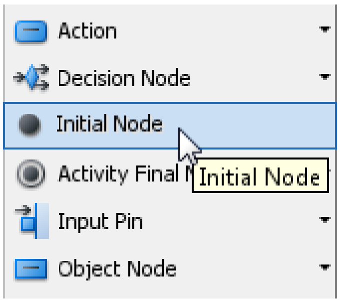


Рисунок 5.27 – Створення Initial Node

2) Створити дію за допомогою значка Control Flow -> Action на Initial Node (рис. 5.28). Створити дію.

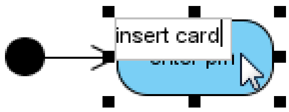


Рисунок 5.28 – Створення контрольного потоку та дії

3) Завершити діаграму фінальним вузлом (рис. 5.29).

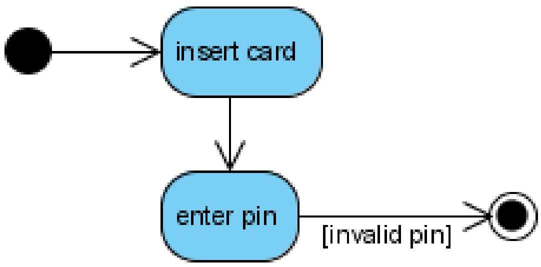


Рисунок 5.29 – Додавання фінального вузла

4) Для того щоб згрупувати дії за учасниками, можна вибрати Horizontal Swimlane (рис. 5.30).

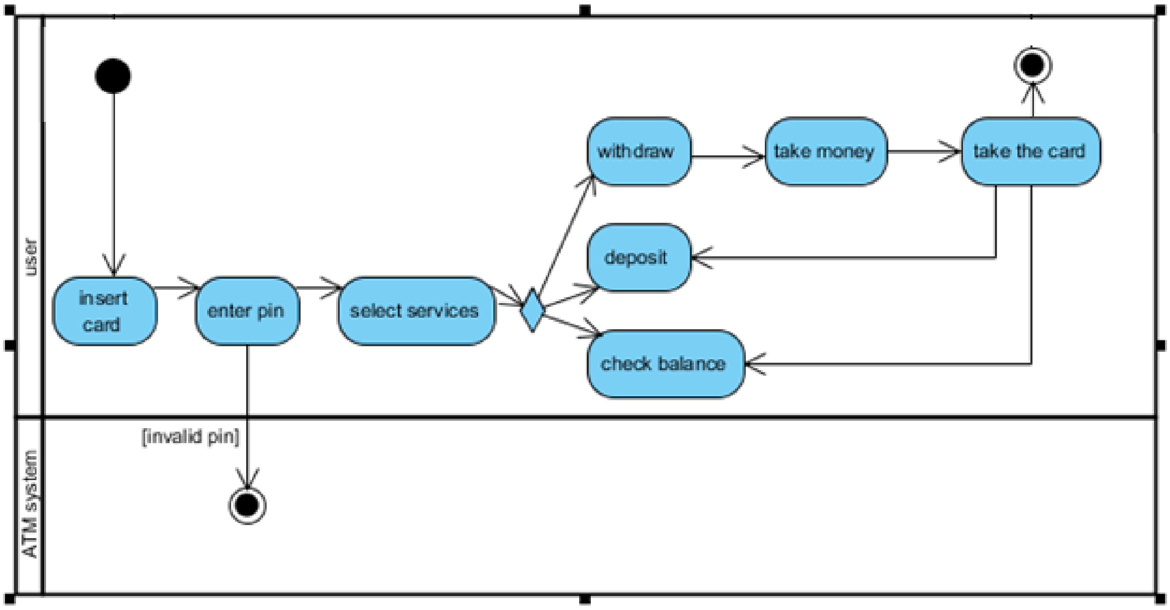


Рисунок 5.30 – Приклад використання Horizontal Swimlane

Діаграма класів відображає об’єкти системи та взаємозв’язки між ними. Щоб створити нову діаграму класів, треба натиснути на Class Diagram правою кнопкою миші у Diagram Navigator та вибрати New Class Diagram. Далі слід виконати послідовність кроків.

1) Вибрати Class у панелі інструментів діаграми та перенести його в область діаграм (рис. 5.31).

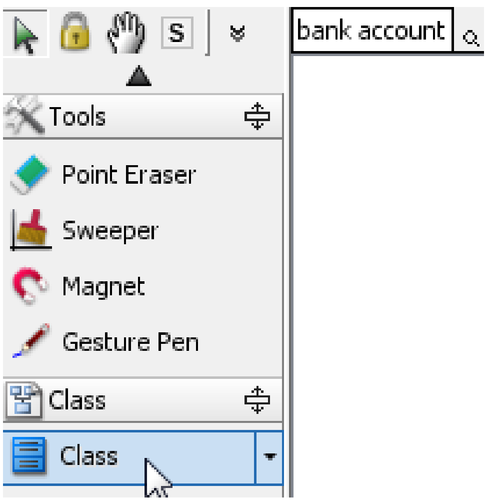


Рисунок 5.31 – Створення класу

2) Назвати клас. Створити атрибути, натиснувши правою кнопкою миші на класі та вибравши Add > Attribute (рис. 5.32). Можна створювати стільки атрибутів, скільки треба. У такий саме спосіб можна створити операції, вибравши Add > Operation.

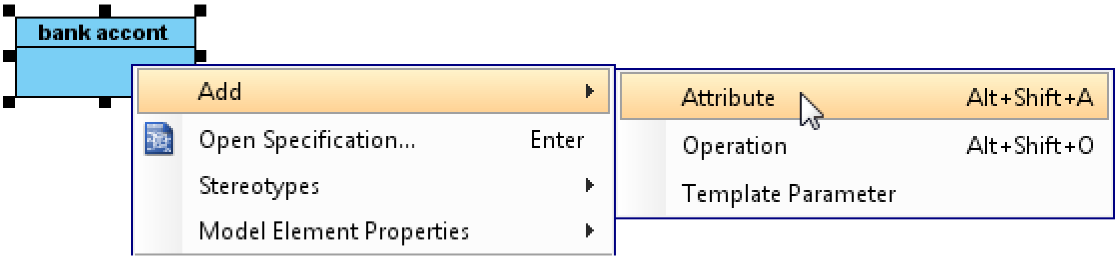


Рисунок 5.32 – Створення атрибутів класу

3) Відношення, наприклад, узагальнення, створюється за допомогою значка Generalization -> Class. Далі створюються підкласи (рис. 5.33).

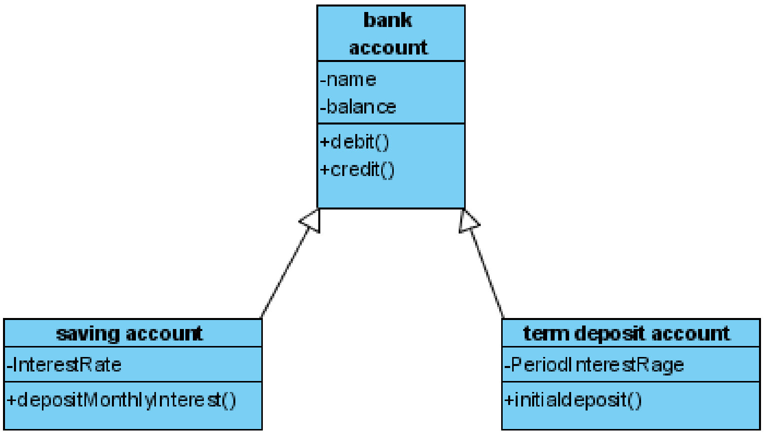


Рисунок 5.33 – Створення відношення узагальнення

Розглянемо приклад розроблення кількох типів UML діаграм у середовищі Visual Paradigm for UML.

Необхідно створити діаграму прецедентів, класів, активності та компонентів для документування розроблення програмного забезпечення для роботи системи обробки замовлень. Замовник може робити два типи замовлень, що обробляються системою: звичайні (normal) та спеціальні (special).

Частина функціональності системи обробки замовлень, обмежена даним описом, може бути зображена на діаграмі прецедентів (рис. 5.34).

Діаграма класів виглядатиме, як показано на рис. 5.35.

Діаграма діяльності зображена на рис. 5.36.

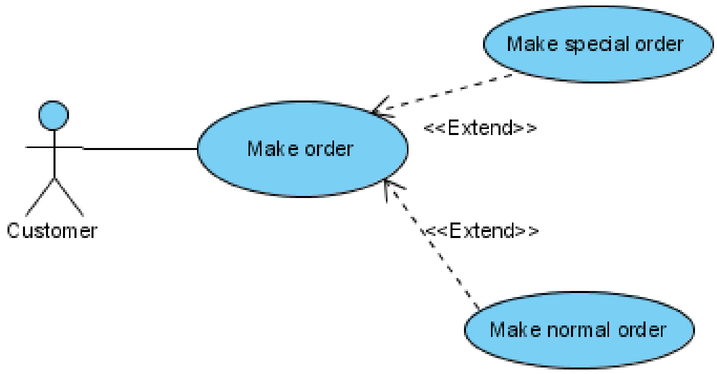


Рисунок 5.34 – Діаграма прецедентів

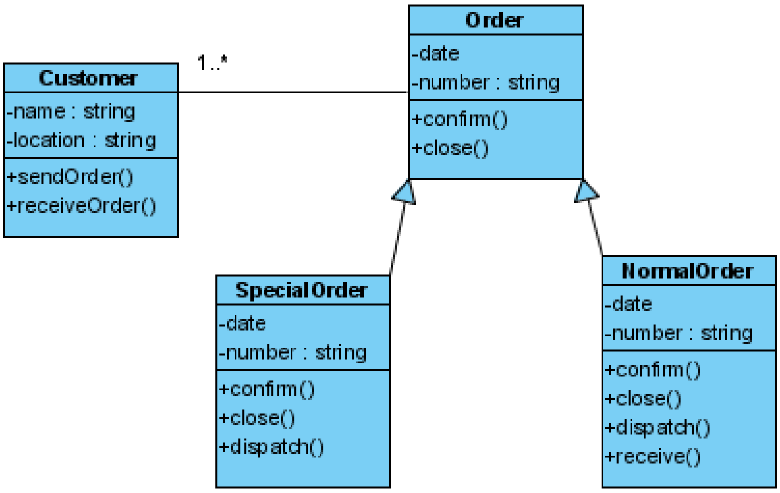


Рисунок 5.35 – Діаграма класів

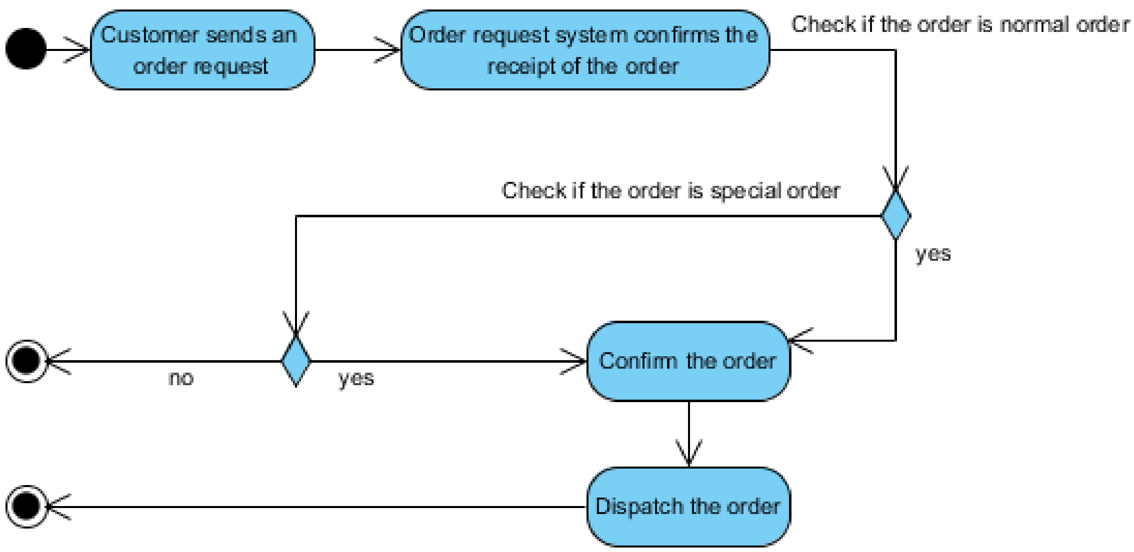


Рисунок 5.36 – Діаграма діяльності

На рис. 5.37 показана діаграма компонентів.

Для того щоб зберегти результати своєї роботи, необхідно вибрати File > Save Project або File > Save Project as…. Зберігаючи вперше, можна обрати місце, куди зберегти проект: до робочої області або в іншу директорію, як показано на рис. 5.38.

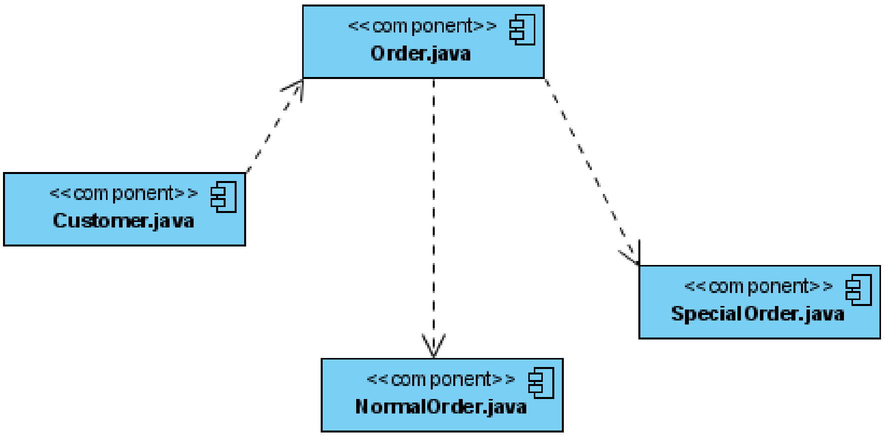


Рисунок 5.37 – Діаграма компонентів

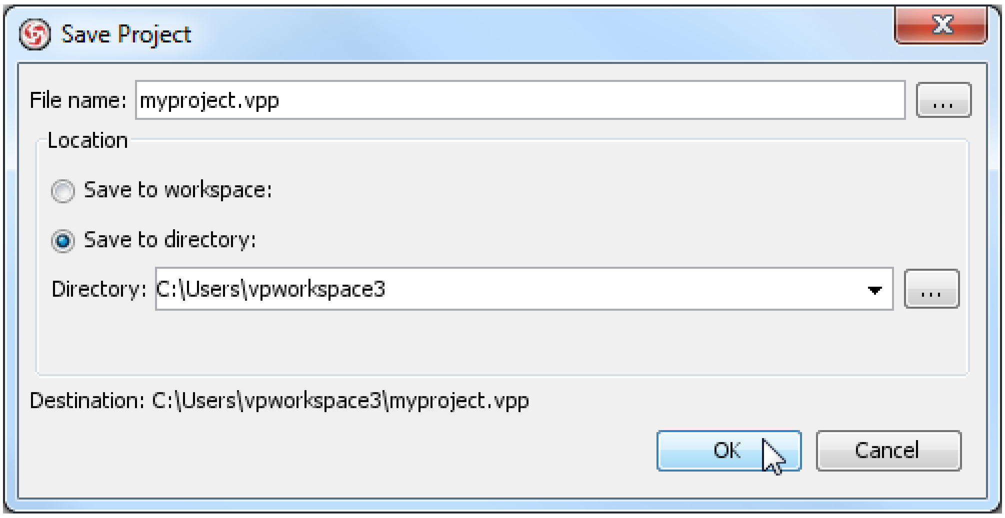


Рисунок 5.38 – Збереження проекту

5.8. Приклад моделювання бізнес-процесу

Як приклад розглянемо процес вивчення деякої дисципліни у ВНЗ. План навчання з кожної дисципліни передбачає розбиття досліджуваного матеріалу на кілька модулів. Кожен модуль закінчується деякою формою контролю (наприклад, контрольною роботою). У рамках вивчення дисципліни передбачено дві основні форми освоєння матеріалу: аудиторні заняття (лекції, практичні заняття, лабораторні роботи) та самостійна робота. Крім цього, з метою закріплення теоретичних знань і практичних навичок можливе виконання індивідуального домашнього завдання (ІДЗ). Як правило, в кінці вивчення дисципліни передбачений підсумковий семестровий контроль у формі іспиту або заліку.

Припустимо, що вивчення даної дисципліни включає відвідування студентами лекцій, виконання лабораторних робіт і самостійну роботу над ІДЗ. Увесь матеріал розбитий на два модулі, а в якості модульного контролю використовується письмова контрольна робота. У кінці семестру студент здає іспит.

Побудуємо контекстну діаграму в IDEF0 (рис. 5.39).

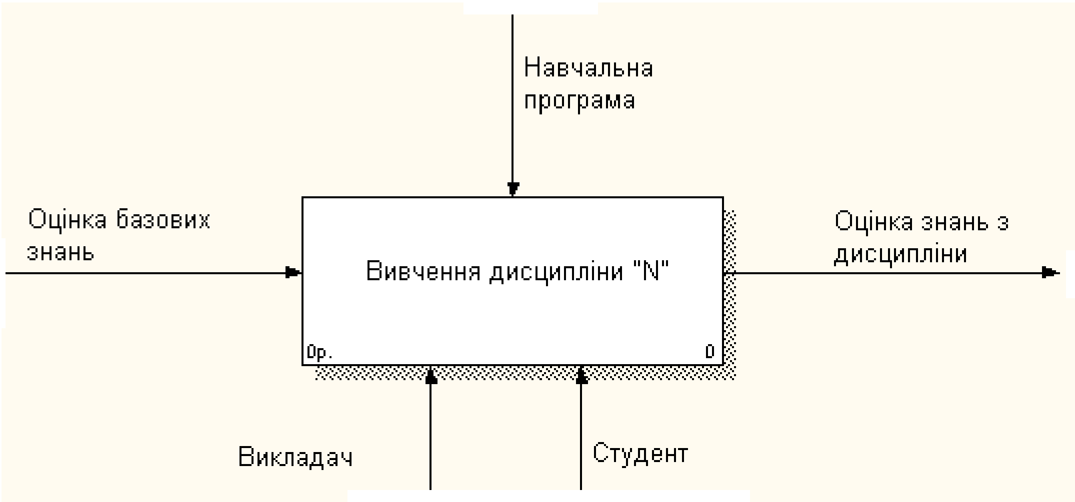


Рисунок 5.39 – Контекстна діаграма вивчення дисципліни

Вивчення кожної дисципліни передбачає, що студент вже має певні знання, необхідні для освоєння нового матеріалу. Вивчення дисципліни регламентується навчальною програмою, яка визначає кількість годин, теми, що розкриваються викладачем, форми контролю. У результаті навчання студент здобуває знання, які оцінюються викладачем.

У ході вивчення дисципліни студент освоює теоретичний матеріал, виконує лабораторні роботи і займається самостійно (рис. 5.40). Поточний контроль передбачає врахування здачі всіх робіт та отримання допуску до іспиту. Підсумковий контроль полягає в проведенні іспиту та оцінки знань з дисципліни.

Самостійна робота (рис. 5.41) передбачає отримання завдання, виконання студентом ІДЗ, а також проведення викладачем консультацій у певний час. Завдання видається згідно з номером студента у списку групи. Для виконання ІДЗ студент користується методичкою.

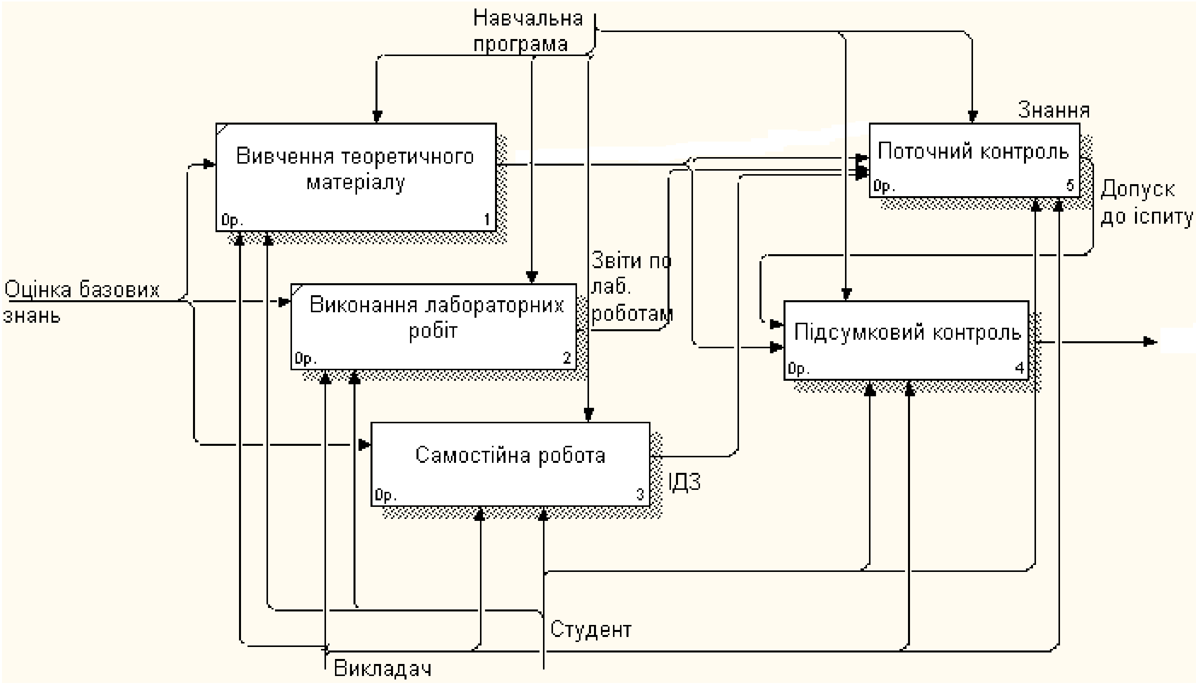


Рисунок 5.40 – Декомпозиція процесу вивчення дисципліни в IDEF0

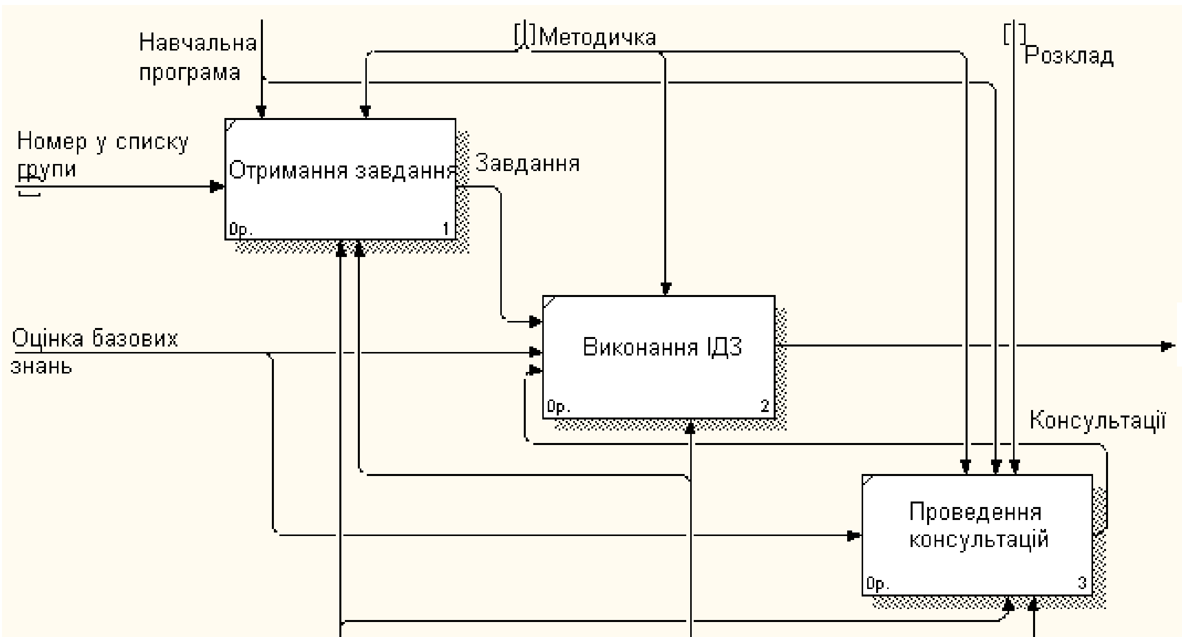


Рисунок 5.41 – Декомпозиція самостійної роботи в IDEF0

Поточний контроль полягає у проведенні модульних контрольних робіт, здачі всіх лабораторних робіт та ІДЗ, на підставі чого викладач допускає студента до іспиту (рис. 5.42). Викладач, спираючись на дані про здані модулі, лабораторні роботи та ІДЗ, що зберігаються в журналі, дає студенту допуск до іспиту або список того, що необхідно доздати (рис. 5.43).

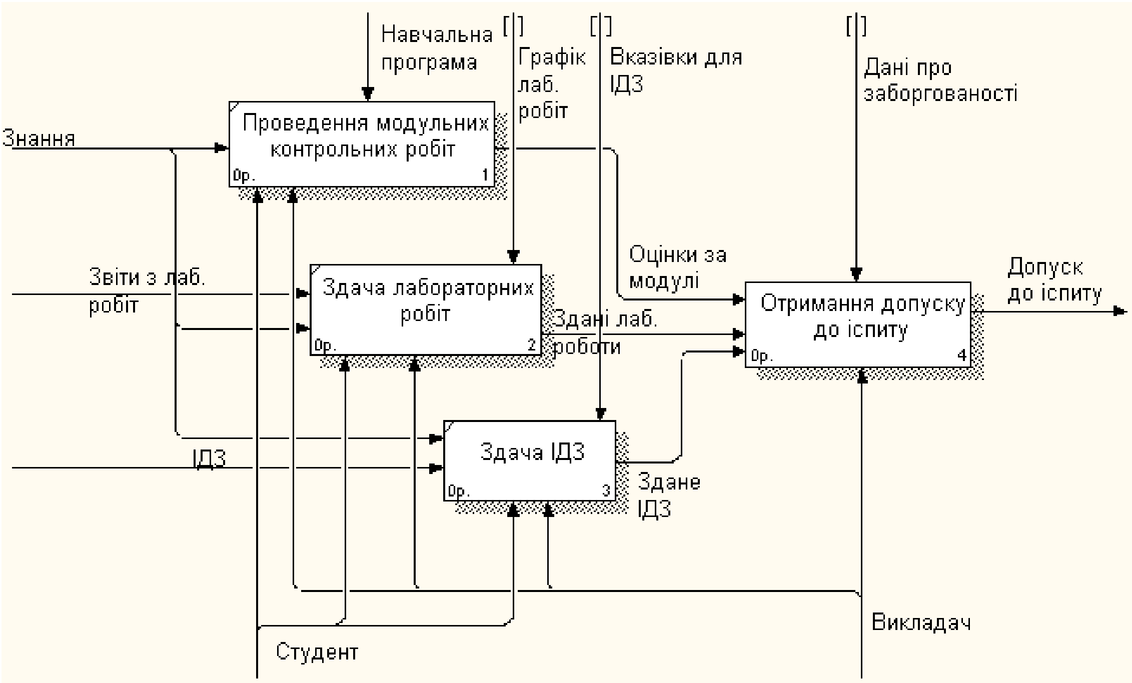


Рисунок 5.42 – Декомпозиція процесу проведення поточного контролю в IDEF0

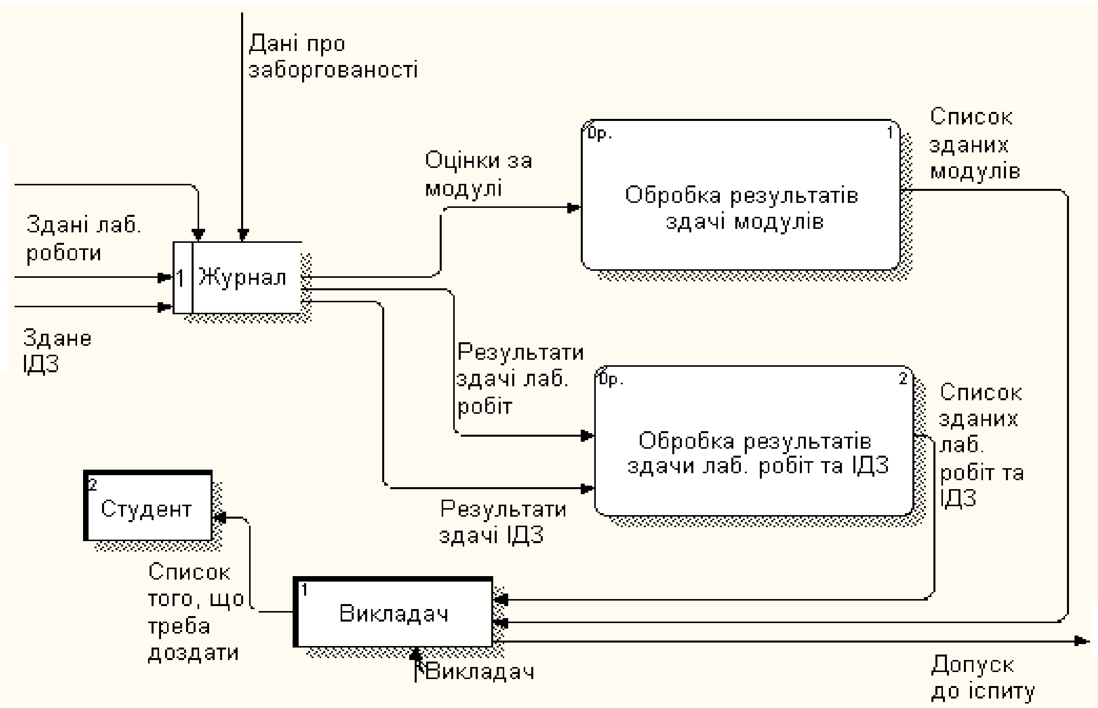


Рисунок 5.43 – Процес отримання допуску до іспиту в DFD

У ході підсумкового контролю визначається оцінка знань студента з дисципліни (рис. 5.44). Вона може відповідати попередньо оцінці, отриманій виходячи з результатів модулів. Бажаючи підвищити оцінку, студент може здавати іспит і отримати підсумкову оцінку.

Описаний процес можна моделювати, використовуючи стандарт UML. У цьому випадку учасники та основні бізнес-функції подаються на діаграмі прецедентів (рис. 5.45).

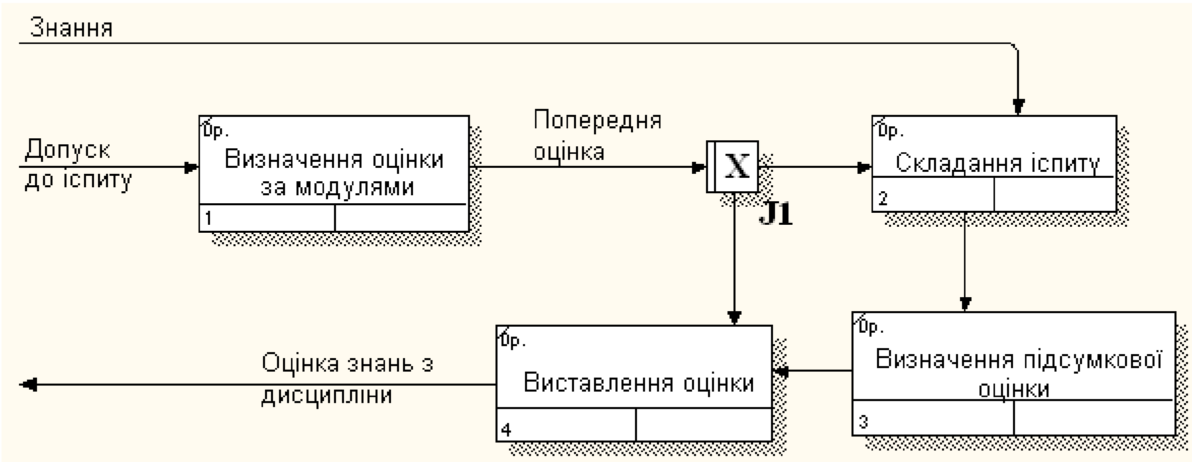


Рисунок 5.44 – Декомпозиція підсумкового контролю в IDEF3

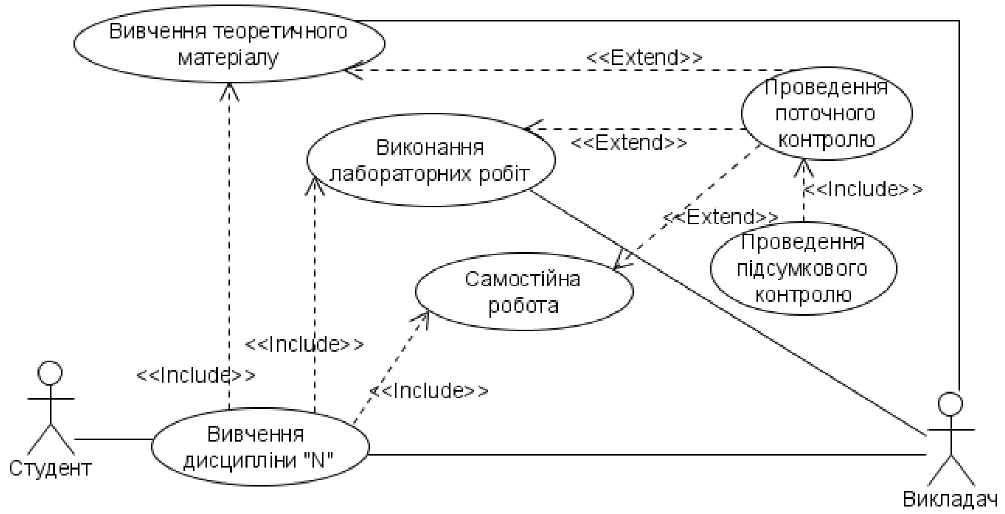


Рисунок 5.45 – Діаграма прецедентів процесу вивчення дисципліни (UML)

Процес самостійної роботи подано у вигляді діаграми діяльності (рис. 5.46). Студент отримує завдання, виконуючи ІДЗ, консультується з викладачем. Завершене ІДЗ здається, викладач перевіряє його. Якщо ІДЗ виконано правильно, це відзначається в журналі, в іншому випадку студент виправляє його і знову здає. Процес отримання допуску до іспиту в нотації UML зображений на рис. 5.47.

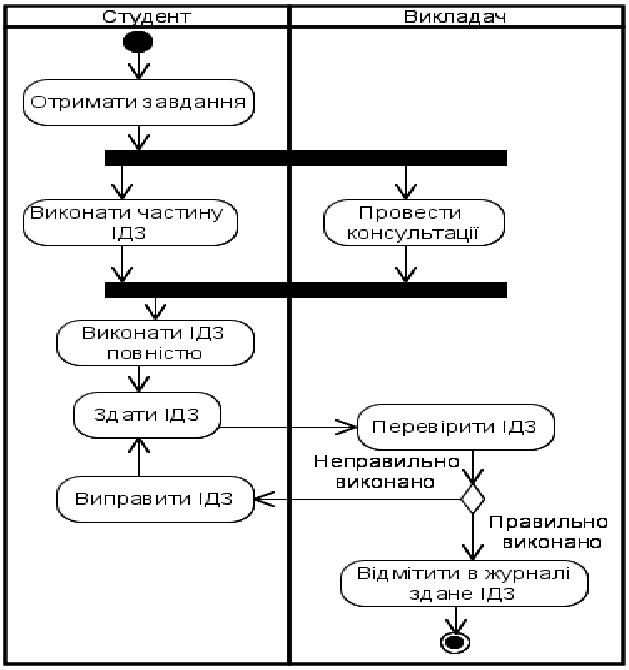


Рисунок 5.46 – Процес здачі ІДЗ (UML)

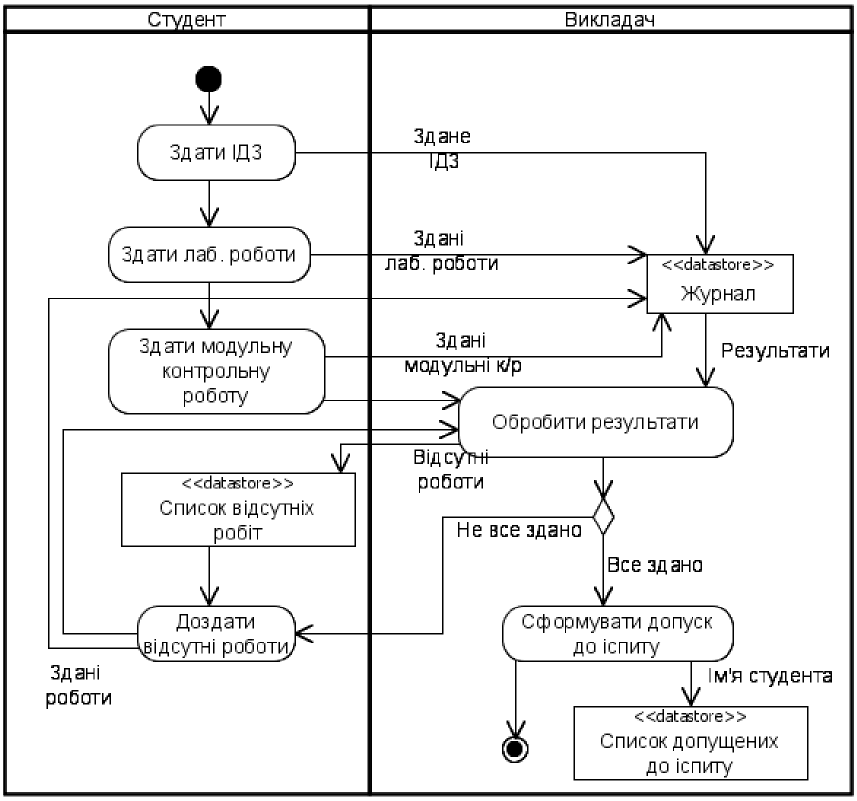


Рисунок 5.47 – Процес отримання допуску до іспиту (UML)

Розглянемо бізнес-процес вивчення дисципліни у ВНЗ у нотації BPMN (рис. 5.48).

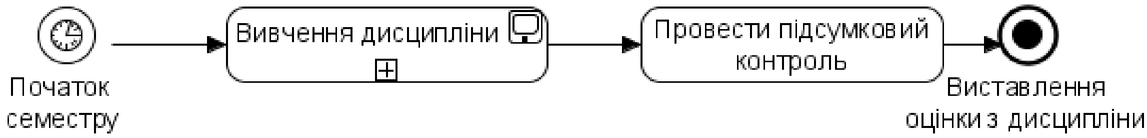


Рисунок 5.48 – Вивчення дисципліни (BPMN)

Підпроцес вивчення дисципліни може бути декомпонованим (рис. 5.49). Він включає вивчення теоретичного матеріалу, виконання лабораторних робіт і ІДЗ, здачу модулів. Результати навчальної діяльності обробляються, і студент отримує допуск до іспиту.

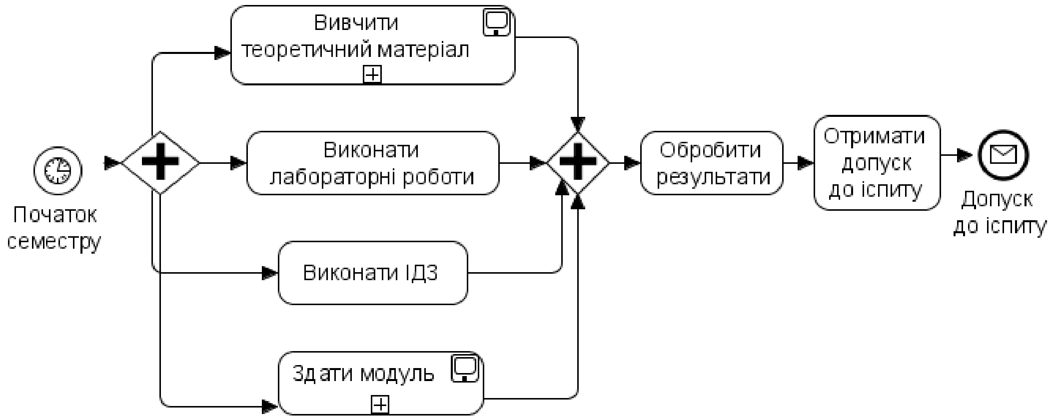


Рисунок 5.49 – Декомпозиція процесу вивчення дисципліни (BPMN)

Вивчення теоретичного матеріалу передбачає відвідування лекцій, що переривається модульним тижнем, а також самостійне вивчення додаткової літератури з дисципліни (рис. 5.50).

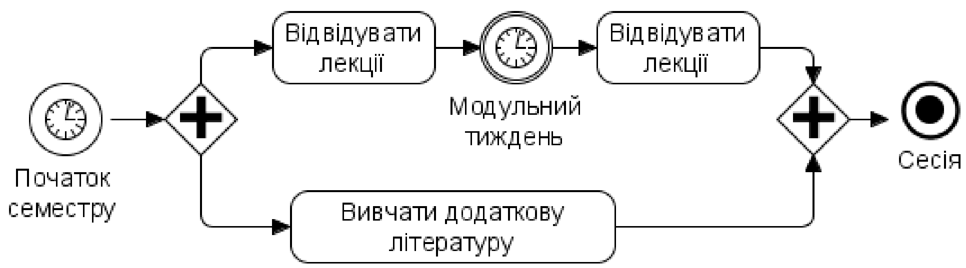


Рисунок 5.50 – Вивчення теоретичного матеріалу (BPMN)

Здача модуля (рис. 5.51) полягає у видачі викладачем завдання для модульної контрольної роботи, її виконанні студентом, перевірці. Якщо викладач не зараховує контрольну роботу, студент перездає її. За зараховану контрольну роботу викладач виставляє оцінку.

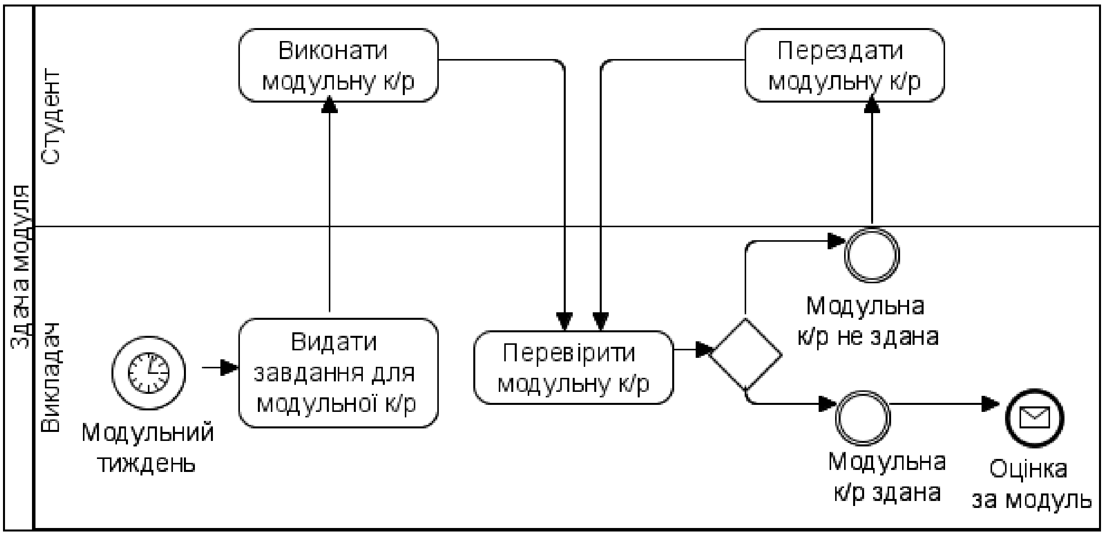


Рисунок 5.51– Здача модуля (BPMN)

Таким чином, розглянутий приклад ілюструє, що для формалізації бізнес-процесів можна використовувати різні підходи і нотації. Вибір методології та CASE-засобів визначається метою і завданнями моделювання, вимогами і обмеженнями процесу моделювання, а також кваліфікацією і вподобаннями аналітика.

Висновки

У даній главі розглянуті проблеми моделювання бізнес-процесів. Проаналізовано основні методології моделювання бізнес-процесів, до яких насамперед належать структурний та об’єктно-орієнтований підходи. Наведено огляд найпоширеніших засобів для візуального моделювання бізнес-процесів, при цьому основна увага приділена CASE-засобу Visual Paradigm for UML. В якості прикладу застосування різних нотацій пропонується моделювання бізнес-процесу вивчення дисципліни.

Завдання

1. Побудуйте моделі бізнес-процесів, які мають місце у супермаркеті, використовуючи нотацію IDEF, DFD, UML та BPMN.
2. Побудуйте моделі бізнес-процесів, які мають місце у компанії, що займається он-лайн продажем залізничних квитків, використовуючи нотацію IDEF, DFD, UML та BPMN.
3. Побудуйте моделі бізнес-процесів, які мають місце у компанії, що займається он-лайн бронюванням готелів, використовуючи нотацію IDEF, DFD, UML та BPMN.
4. Побудуйте моделі бізнес-процесів, які відображають функціонування інтернет-банкінгу, використовуючи нотацію IDEF, DFD, UML та BPMN.
5. Побудуйте моделі бізнес-процесів, які мають місце у сучаній бібліотеці, використовуючи нотацію IDEF, DFD, UML та BPMN.
6. Побудуйте моделі бізнес-процесів, які відображають функціонування інтернет-магазину, використовуючи нотацію IDEF, DFD, UML та BPMN.
7. Побудуйте моделі бізнес-процесів, які відображають функціонування кол-центру, використовуючи нотацію IDEF, DFD, UML та BPMN.
8. Побудуйте моделі бізнес-процесів, які мають місце у компанії, що займається проведенням он-лайн опитувань, використовуючи нотацію IDEF, DFD, UML та BPMN.
9. Побудуйте моделі бізнес-процесів, які мають місце у туристичній компанії, використовуючи нотацію IDEF, DFD, UML та BPMN.
10. Побудуйте моделі бізнес-процесів, які відображають функціонування служби підтримки мобільного оператора, використовуючи нотацію IDEF, DFD, UML та BPMN.

Контрольні питання

1. Що таке бізнес-процес?
2. Яка мета моделювання бізнес-процесів?
3. Які види бізнес-процесів можна виділити?
4. Опишіть переваги та недоліки існуючих CASE-засобів, які використовуються для моделювання бізнес-процесів.
5. Охарактеризуйте можливості нотації IDEF0 для моделювання бізнес-процесів.
6. Для чого використовується нотація DFD?
7. Які можливості надає мова UML для моделювання бізнес-процесів?

8. Для чого призначено середовище Visual Paradigm for UML?
9. Які типи діаграм можна побудувати за допомогою Visual Paradigm for UML?
10. Які нотації візуального моделювання підтримує Visual Paradigm for UML?
11. Назвіть послідовність дій щодо створення діаграми прецедентів.
12. Назвіть послідовність дій щодо створення діаграми класів.
13. Назвіть послідовність дій щодо створення діаграми активності.
14. Назвіть послідовність дій щодо створення діаграми компонентів.
15. Назвіть послідовність дій щодо створення діаграми пакетів.
16. Охарактеризуйте можливості нотації BPMN для моделювання бізнес-процесів.
17. Які типи подій та тригери використовуються в нотації BPMN?

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Макаров И.М. Целевые комплексные программы / И.М. Макаров, В.Б. Соколов, А.А. Абрамов. – М.: «Знание», 1980. – 135 с.
2. Месарович М. Общая теория систем и ее математические основы. Исследования по общей теории систем / И.М. Месарович. – М.: «Прогресс», 1969. – 321 с.
3. Акофф Р. Системы, организации и междисциплинарные исследования. Исследования по общей теории систем / Р. Акофф. – М.: «Прогресс», 1969. – 286 с.
4. Поспелов Г.С. Программно-целевое планирование и управление / Г.С. Поспелов, В.А. Ириков. – М.: «Сов. Радио», 1976. – 440 с.
5. Хорафас Д.Н. Системы и моделирование / Д.Н. Хорафас. – М.: «Мир», 1967. – 389 с.
6. Советов Б.Я. Моделирование систем / Б.Я. Советов, С.А. Яковлев. – М.: Высш. школа, 1985. – 315 с.
7. Цвиркун А.Д. Структура сложных систем / А.Д. Цвиркун. – М.: «Сов. радио», 1975. – 200 с.
8. Норенков И.П. Системы автоматизированного проектирования: учеб. пособие для вузов: В 9 кн. Кн.1. Принципы построения и структура / И.П. Норенков. – Мн.: Высш. шк., 1987. – 123 с.
9. Капустин Н.М. Системы автоматизированного проектирования : учеб. пособие для техн. вузов. В 9 кн. Кн. 6. Автоматизация конструкторского и технологического проектирования / Н.М. Капустин, Г.Н. Васильев. – Мн.: Высш. шк., 1988. – 191 с.
10. Фокс А. Вычислительная геометрия. Применение в проектировании на производстве / А. Фокс, М. Пратт. – М.: Мир, 1982. – 304 с.
11. Завьялов Ю.С. Методы сплайн-функций / Ю.С. Завьялов, Б.И. Квасов, В.Л. Мирошниченко. – М.: Наука, 1980. – 352 с.
12. Ларичев О.И. Качественные методы принятия решений / О.И. Ларичев, Е.М. Мовшович. – М.: Наука, 1996. – 268 с.
13. Растригин Л.А. Современные принципы управления сложными объектами / Л.А. Растригин. – М.: Сов. радио, 1980. – 232 с.

14. Бусленко Н.П. Лекции по теории сложных систем / Н.П. Бусленко, В.В. Калашников, И.Н. Коваленко. – М.: Сов. радио, 1973. – 440 с.
15. Красовский А.А. Системы автоматического управления полетом и их аналитическое конструирование / А.А. Красовский. – М.: Наука, 1973. – 356 с.
16. Первозванский А.А. Математическое модели в управлении производством / А.А. Первозванский. – М.: Наука, 1975. – 616 с.
17. Поспелов Д.А. Логические методы анализа и синтеза схем / Д.А. Поспелов. – М.: Энергия, 1968. – 318 с.
18. Закревский А.Д. Алгоритмы синтеза дискретных автоматов / А.Д. Закревский. – М.: Наука, 1971. – 450 с.
19. Вентцель Е.С. Теория вероятностей / Е.С. Вентцель. – М.: Наука, 1969. – 580 с.
20. Гамаюн И.П. Разработка имитационных моделей на основе сетей Петри : учеб. пособие / И.П. Гамаюн. – Харьков: НТУ «ХПИ», 2002. – 143 с.
21. Гнеденко Б.В. Введение в теорию массового обслуживания / Б.В. Гнеденко, И.Н. Коваленко. – М.: Наука, 1966. – 378 с.
22. Бусленко В.Н. Автоматизация имитационного моделирования сложных систем / В.Н. Бусленко. – М.: Наука, 1977. – 240 с.
23. Михалевич В.С. Вычислительные методы исследования и проектирования сложных систем / В.С. Михалевич, В.Л. Волкович. – М.: Наука, 1982. – 286 с.
24. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации. – К.: Наукова думка, 1988. – 472 с.
25. Томашевский В.Н., Жданова Е.Г. Имитационное моделирование в среде GPSS. – М.: Бестселлер, 2003. – 416 с.
26. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 264 с.
27. Буч Г. Язык UML. Руководство пользователя / Г. Буч, Дж. Рамбо, А. Якобсон. – М.: ДМК Пресс, 2007. – 496 с.
28. Буч Г. UML 2-е издание / Г. Буч, А. Якобсон, Дж. Рамбо. – Санкт-Питербург: Питер, 2006. – 736 с.
29. XML Process Definition Language // http://www.xpdl.org.
30. BPEL: Who Needs It Anyway? // http://www.bpm.com/bpel-who-needs-it.html.
31. Бизнес-процесс //http://www.re-business.ru/standarts/bus_proc /index.html.
32. Основы формальных методов описания бизнес-процессов : учеб. пособие / К.Е. Самуйлов, Н.В. Серебренникова, А.В. Чукарин, Н.В. Яркина. – М.: РУДН, 2008. – 130 с.

33. Маклаков С.В. Моделирование бизнес-процессов с BPWin 4.0 / С.В. Маклаков. – М.: Диалог-МИФИ, 2002. – 209 с.
34. Вендров А.М. Современные технологии создания программного обеспечения (обзор) / А.М. Вендров // Jet Info Информационный бюллетень. – 2004. – №4(131).– с. 2–32.
35. Черемных С.В. Моделирование и анализ систем. IDEF-технологии: практикум / С.В. Черемных, И.О. Семенов, В.С. Ручкин. – М.: Финансы и статистика, 2006. – 192 с.
36. Business Modeling with UML: Business Patterns at Work. – John Wiley & Sons, 2000. – 459 p.
37. W.M.P. van der Aalst. Formalization and Verification of Event-driven Process chains // http://www.alexandria.tue.nl/extra1/wskrap/publichtml/9714860.pdf.
38. Business Process Model and Notation (BPMN). Version 2.0 // http://www.omg.org/spec/BPMN/2.0.
39. Войнов Н. Практика использования BPMN / Н. Войнов // http://nvoynov.blogspot.com/2007/10/bpmn.html.
40. Assaf Arkin. Business Process Modeling Language // http://www.bpmn.org/Documents/BPML-2003.pdf.

Навчальне видання

ГАМАЮН Ігор Петрович
ЧЕРЕДНІЧЕНКО Ольга Юріївна

МОДЕЛЮВАННЯ СИСТЕМ

Навчальний посібник для студентів спеціальностей
6.050103 “Програмна інженерія”
6.050101 “Комп’ютерні науки”

Відповідальний за випуск *М. Д. Годлевський*

Роботу до видання рекомендував *О. В. Горілий*

Редактор *О. І. Шпільова*

Формат 60×84/16. Ум. друк. арк. 13,25. Наклад 300 прим. Зам. № 20_07-1.

Видавництво «Факт»
Україна, 61166, м. Харків, вул. Бакуліна, 11, оф. 4-28.
Тел./факс: (057)756-43-75. E-mail: publish_fakt@mail.ru
Свідоцтво суб’єкта видавничої справи ДК № 3172 від 22.04.2008

Виготовлено у ФОП В. Є. Гудзинський
Україна, 61072, м. Харків, вул. 23-го Серпня, 27.
Тел./факс: (057)340-52-26. E-mail: fop_veg@ukr.net
Свідоцтво суб’єкта видавничої справи ХК № 269 від 23.11.2010