

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра комп'ютерних наук

ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ
із дисципліни

**“МОДЕЛЮВАННЯ ТА АНАЛІЗ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ”**

Тернопіль – ЗУНУ-2024

Опорний конспект лекцій із дисципліни «Моделювання та аналіз програмного забезпечення» для здобувачів освітнього рівня «бакалавр» спеціальності 121 – Інженерія програмного забезпечення галузі знань 12 – Інформаційні технології // Л.І. Гончар, Є.О.Марценюк .– Тернопіль: ЗУНУ, 2024. – 63 с.

Укладачі: **Гончар Людмила Іванівна**, к.е.н., доцент кафедри комп'ютерних наук ТНЕУ
Марценюк Євгенія Олексіївна, к.т.н., доцент кафедри комп'ютерних наук ТНЕУ

Відповідальний за випуск: **Пукас Андрій Васильович**, д.т.н., професор, завідувач кафедри комп'ютерних наук ТНЕУ

Рецензенти: **Якименко Ігор Зіновійович**, к.т.н., доцент кафедри кібербезпеки ЗУНУ, декан ФКІТ
Франко Юрій Павлович, к.т.н., завідувач кафедри комп'ютерних технологій Тернопільського національного педагогічного університету ім. В. Гнатюка

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри комп'ютерних наук ЗУНУ (протокол № 1 від 27 серпня 2024 року)

© Гончар Л. І.,
Марценюк Є. О., 2024
© ЗУНУ, 2024

ЗМІСТ

Тема 1. Загальні положення та визначення	4
Тема 2. Моделі систем масового обслуговування	23
Тема 3. Імітаційне моделювання	35
Тема 4. Метод Монте-Карло	47
Тема 5. Основи моделювання подій	52
Тема 6. Основи моделювання архітектури ПЗ	56
Рекомендовані джерела інформації	62

ТЕМА 1

ЗАГАЛЬНІ ПОЛОЖЕННЯ ТА ВИЗНАЧЕННЯ:

- 1.1. Поняття системи
 - 1.2. Поняття моделі
 - 1.3. Співвідношення між моделлю та системою
 - 1.4. Класифікація моделей
 - 1.5. Вимоги до моделей
 - 1.6. Основні види моделювання
 - 1.7. Формальні методи побудови моделей
 - 1.8. Кібернетичний підхід
 - 1.9. Теоретико-множинний підхід
 - 1.10. Принципи побудови моделей
 - 1.11. Технологія моделювання: основні етапи, їх взаємозв'язок та характеристики
- Висновки**
- Контрольні запитання**

Моделювання – це спосіб дослідження будь-яких явищ, процесів або об'єктів шляхом побудови та аналізу їх моделей. У широкому розумінні моделювання є однією з основних категорій теорії пізнання і чи не єдиним науково обґрунтованим методом наукових досліджень систем і процесів будь-якої природи в багатьох сферах людської діяльності.

У цьому розділі розглядаються основні відомості загальної теорії систем і моделювання.

1.1. Поняття системи

Основними поняттями в теорії і практиці моделювання об'єктів, процесів і явищ є «система» та «модель».

У перекладі з грецької «systema» – ціле, яке складається із частин; об'єднання. Термін «система» існує вже більш ніж два тисячоліття, проте різні дослідники визначають його по-різному. На сьогодні існує понад 500 визначень терміну «система». Однак, використовуючи будь-яке з них, у першу чергу потрібно мати на увазі ті завдання, які ставить перед собою дослідник. Системою може бути і один комп'ютер, і автоматизована лінія або технологічний процес, в яких комп'ютер є лише одним із компонентів, і все підприємство або кілька різних підприємств, які функціонують як єдина система в одній галузі промисловості. Те, що один дослідник визначає як систему, для іншого може бути лише компонентом більш складної системи.

Для всіх визначень системи загальним є те, що система – це цілісний комплекс взаємопов'язаних елементів, який має певну структуру і взаємодії із зовнішнім середовищем. *Структура* системи – це організована сукупність зв'язків між її елементами. Під таким зв'язком розуміють можливість впливу одного елемента системи на інший. *Середовище* – це сукупність елементів зовнішнього

світу, які не входять до складу системи, але впливають на її поведінку або властивості. Система є *відкритою*, якщо існує зовнішнє середовище, яке впливає на систему, і *закритою*, якщо воно відсутнє або з огляду на мету досліджень не враховується.

Одне з перших визначень системи належить американському біологу Л. фон Берталанфі, згідно з яким система складається з деякої кількості взаємопов'язаних елементів. Оскільки між елементами системи існують певні взаємозв'язки, то мають бути структурні відношення. Таким чином, система – це щось більше, ніж сукупність елементів. Аналізуючи систему, потрібно враховувати оцінку системного (синергетичного) ефекту. Властивості системи відмінні від властивостей її елементів, і залежно від властивостей, якими цікавляться дослідники, та ж сама сукупність елементів може бути системою або ні.

Багато дослідників визначають систему як *цілеспрямовану* множину взаємопов'язаних елементів будь-якої природи. Згідно з цим визначенням система функціонує для досягнення деякої мети. Це визначення є правильним для соціологічних і технічних систем, але не підходить для систем навколишньої природи (наприклад, біологічних), мета функціонування яких не завжди відома.

Одне з важливих визначень системи пов'язане з абстрактною теорією систем, у рамках якої, на відміну від Інших рівнів опису систем, використовуються такі рівні абстрактного опису:

- ◆символічний, або лінгвістичний;
- ◆теоретико-множинний;
- ◆абстрактно-алгебричний;
- ◆топологічний;
- ◆логіко-математичний;
- ◆теоретико-інформаційний;
- ◆динамічний;
- ◆евристичний.

Найвищий рівень абстрактного опису систем – *лінгвістичний*; ґрунтуючись на ньому, можна одержати всі інші рівні. На цьому рівні вводиться поняття предметної області, для опису якої застосовуються алгебричні моделі, з якими пов'язана деяка мова. Для опису предметної області цією мовою використовуються два рівні формальних мов, за допомогою яких будують логіко-алгебричну модель предметної області. На цій моделі підтверджуються методи дослідження за допомогою формального апарату, яким можуть бути теорії, побудовані у вигляді істинних висловлювань з усієї множини висловлювань.

Таким чином, система - це окремий випадок теорії, описаний формальною мовою, яка уточнюється до мови об'єктів. Для визначення деякого поняття використовують певні символи (алфавіт) і встановлюють правила оперування ними. Сукупність символів і правил користування ними утворює абстрактну мову. Поняття, висловлене абстрактною мовою, означає будь-яке речення (формулу), побудоване за граматичними правилами цієї мови. Припускають,

що таке речення містить варійовані змінні, так звані *конституенти*, які, маючи тільки певні значення, роблять дане висловлювання істинним.

Таким чином, система S задається парою елементів:

$$S = \langle X_s, R_s \rangle,$$

де X_s, R_s – множини відповідно елементів і відношень між ними. Відношення визначають взаємодію між об'єктами. У загальному випадку n - відношення R^n у множинах, є деякою підмножиною декартового добутку $X_1 \times X_2 \times \dots \times X_n$, який зіставлено з n -вимірних наборів (кортежів) виду (x_1, x_2, \dots, x_n) , де $x_i \in X_i$,
 $i = 1, 2, \dots, n$.

Якщо відношення R^n в окремому випадку задається, наприклад, деякою функцією, що визначає зв'язок між певним елементом $x \in X$ і певною підмножиною Y :

то $f: X \rightarrow Y$, тобто вважаємо, що функція f перетворює значення із множини X у значення підмножини Y . Для функції f множина X – це область визначення, а підмножина Y – область значень функції. Функцію f можна подати як множину впорядкованих пар елементів (x, y)

Що стосується атрибутів системи, то вони подібні до функцій, визначених у підмножині об'єктів. Відмінність атрибутів від функцій полягає в тому, що два різних атрибути з погляду на поняття функції можуть бути однаковими. Атрибут A задається парою елементів – (i, f) , де i – ім'я атрибута, а f – функція, визначена на підмножині об'єктів. У динамічних об'єктів атрибут також може бути функцією від часу t .

Наприклад, у разі дослідження пропускнуої здатності ділянок доріг об'єктами системи можуть бути перехрестя, розв'язки, поворот і прямолінійні ділянки доріг (статичні об'єкти) та автомобілі (динамічні об'єкти). Властивості (атрибути) динамічних об'єктів, на відміну від властивостей статичних, змінюються в часі. Наприклад, гальмовий шлях автомобіля змінюється залежно від швидкості руху та погодних умов, а прискорення може бути додатним (під час розгону) або від'ємним (під час гальмування). Відношення в цій системі задаються згідно з правилами дорожнього руху.

Вивчаючи систему більш глибоко, усвідомлюємо, що вона може складатися з підсистем або бути однією з елементів більшої системи, тобто може існувати ієрархія систем. Наприклад, двигун є підсистемою автомобіля, який у свою чергу є підсистемою транспортного потоку магістралі.

На теоретико-множинному рівні абстрактного опису системи можна отримувати досить загальні відомості про реальні системи, а для конкретніших цілей потрібні інші моделі, які давали б змогу більш детально аналізувати різні властивості реальних систем. Для цього потрібні нижчі рівні абстрактного опису систем, які є окремими випадками опису теоретико-множинного рівня. Так, якщо зв'язки між елементами розглядуваних множин устанавлюються за допомогою деяких однозначних функцій, що відображають елементи множини в саму відповідну множину, то має місце *абстрактно-алгебричний* рівень опису систем..

Якщо ж на множинах, які розглядаються, визначено деякі багатозначні функції, то мають місце *топологічні абстрактні* моделі, записані мовою

загальної топології або її гілок, які називаються гомологічною топологією, алгебричною топологією тощо. Вибір потрібного рівня абстрактного опису під час вивчення тієї або іншої реальної системи є завжди найвідповідальнішим і найважчим кроком у теоретико-системних побудовах. Цей процес майже не піддається формалізації і багато в чому залежить від досвіду та знань дослідника, його фахової належності, цілей дослідження тощо.

Можна показати, як від систем із топологічним рівнем опису перейти до узагальнених динамічних. Щоб дати строге математичне визначення поняттю *динамічна система*, її наділяють властивістю мати «входи» й «виходи», тобто визначають як

структурований об'єкт, куди в певні моменти часу можна вводити речовину, енергію та інформацію, а в інші моменти – виводити їх. Динамічні системи можна зобразити і як системи, де процеси відбуваються неперервно, і як системи, в яких усі процеси протікають лише в дискретні моменти часу.

Інші абстрактні рівні опису систем пов'язані з розвитком інформаційних і програмних систем, а також систем штучного інтелекту.

Елементи системи і зв'язки між ними в різних випадках можуть мати різну природу (фізичну, інформаційну, технологічну, біологічну, соціальну), тому аналізом систем займаються представники різних галузей науки і техніки. Науковий напрям під назвою загальна теорія систем, який з'явився наприкінці 50-х – на початку 60-х років ХХ сторіччя, пов'язаний із розробленням сукупності філософських, методологічних, наукових і прикладних методів аналізу та синтезу систем довільної природи. Ця теорія є загальною, оскільки має дедуктивний характер, об'єднує інші теорії, а саме: теорії керування, самоорганізації, навчання тощо, і розроблена для вивчення поведінки абстрактних систем. Основне її призначення – пояснити, яким чином з окремих елементів утворюється складна єдність цілого, нова сутність. Загальна теорія систем тісно пов'язана з формальною і є певною мірою математичною. Основна процедура теорії систем і системного аналізу – *побудова моделі* системи, яка відображала б усі фактори, взаємозв'язки і реальні ситуації. Займаються цим спеціалісти із системного аналізу – системотехніки або системні аналітики.

1.2. Поняття моделі

Науковою основою моделювання як методу пізнання і дослідження різних об'єктів і процесів є *теорія подібності*, в якій головним є поняття *аналогії*, тобто схожості об'єктів за деякими ознаками. Подібні об'єкти називаються аналогами. Аналогія між об'єктами може встановлюватись за якісними і (або) кількісними ознаками.

Основним видом кількісної аналогії є *математична подібність*, коли об'єкти описуються за допомогою рівнянь і функцій. Функції та незалежні змінні називаються схожими, якщо вони співпадають з точністю до деякої константи. Окремими видами математичної подібності є *геометрична подібність*, яка встановлює подібність геометричних образів, і *часова*, що визначає подібність функції часу, для якої константа часу (масштаб) показує, в яких відношеннях знаходяться параметри функцій, такі як період, часова затримка тощо.

Іншим видом кількісної аналогії, який слід відзначити, є *фізична подібність*. Критерії фізичної подібності можна отримати, не маючи математичного опису об'єктів, наприклад на основі значень фізичних параметрів, які характеризують досліджуваний процес у природі й на моделі. За типом процесу розрізняють види подібності, для яких розроблено відповідні критерії – гідравлічні, електричні, аеродинамічні та ін.

Вивчення переходу від властивостей реальних об'єктів до властивостей системи є найважливішим завданням теорії систем. У загальній теорії систем визнається об'єктивність їх існування. Згідно з цією теорією, якщо реально існують взаємозв'язки між об'єктами, то існують і системи, які їм відповідають. Ця теорія ґрунтується на постулаті функціонально-структурного ізоморфізму об'єктів і явищ природи.

Якщо структура однієї системи і зовнішні функції її елементів ізоморфні структурі іншої системи і зовнішнім функціям її елементів, то зовнішні властивості цих систем не розрізняються в області їх ізоморфізму. У теорії систем цей постулат має не менше значення, ніж закони збереження матерії у фізиці або аксіоми в математиці. Разом з іншими постулатами він є підґрунтям для логічного, доказового розгортання теорії і дає можливість пояснити єдність закономірностей природи для об'єктів, які здаються несхожими і незалежними один від одного. Ізоморфізм реальних систем є основою і логічним наслідком вищезазначеного постулату.

У теорії систем існує ще один важливий для моделювання постулат, який визначає, що описом структури і функцій деякої системи може бути інша ізоморфна стосовно неї система. Ця ізоморфність (подібність) двох систем стосується структур систем і функцій їх елементів. Одна з таких систем є *моделлю* іншої (*оригіналу*) і навпаки. Таких ізоморфних систем може бути безліч. Виникає проблема вибору або побудови системи, яка може бути моделлю досліджуваної системи.

Теорія подібності дає змогу встановити відношення еквівалентності (відповідності, схожості) між двома розглядуваними системами за деякими ознаками. Будь-яка з цих систем може існувати реально або бути абстрактною. Якщо система існує реально, то її можна вивчати, досліджуючи, яким чином пов'язані вхідні впливи з виходами системи. На основі результатів досліджень будується деяка *абстрактна система*, де відношення еквівалентності визначають тільки ті істотні властивості та аспекти поведінки, які у вихідній та абстрактній системах мають бути однаковими. М. Месарович відзначає, що, базуючись на спостереженнях і дослідженнях однієї системи, можна робити висновки про властивості та поведінку іншої. Здебільшого на практиці абстрактна система *простіша* за вихідну, якщо не враховувати тих аспектів, що визначають відношення еквівалентності.

Таким чином, можна перейти до визначення терміна «*модель*». У філософській літературі терміном «*модель*» позначають «*деяку реально існуючу систему або ту, що представляється в думках, яка, заміщаючи і відображаючи в пізнавальних процесах іншу систему-оригінал, знаходиться з нею у відношенні схожості (подібності), завдяки чому вивчення моделі дає змогу отримати нову інформацію про оригінал*». У цьому визначенні закладено

генетичний зв'язок моделювання з теорією подібності, принципом аналогії. Таким чином, *моделлю* можна називати систему, яку використовують для дослідження.

Термін «модель» походить від латинського слова «*modulus*», тобто зразок, пристрій, еталон. У широкому значенні – це будь-який аналог (уявний, умовний: зображення, опис, схема, креслення тощо) певного об'єкта, процесу, явища («оригіналу» даної моделі), що використовується як його «замінник». Цей термін можна застосовувати також для позначення системи постулатів, даних і доведень, формального опису деякого явища або стану речей. У сучасній теорії керування використовуються моделі двох основних типів. Для технологічних об'єктів цей поділ відповідає «феноменологічним» і «дедуктивним» моделям. Під феноменологічними моделями розуміють переважно емпірично поновлені залежності вихідних даних від вхідних, як правило, з невеликою кількістю входів і виходів. Дедуктивне моделювання передбачає з'ясування та опис основних фізичних закономірностей функціонування всіх компонентів досліджуваного процесу і механізмів їх взаємодії. За допомогою дедуктивних моделей описується процес у цілому, а не окремі його режими.

Перший тип моделей – *моделі даних*, які не потребують, не використовують і не відображають будь-яких гіпотез про фізичні процеси або системи, з яких ці дані отримано. До моделей даних належать усі моделі математичної статистики. Останнім часом ця сфера моделювання пов'язується з експериментально - статистичними методами і системами, що істотно розширює методологічну базу для прийняття рішень під час розв'язання завдань аналізу даних і керування.

Другий тип моделей – *системні моделі*, які будуються в основному на базі фізичних законів і гіпотез про те, як система структурована і, можливо, як вона функціонує. Використання системних моделей передбачає можливість працювати в технологіях віртуального моделювання – на різноманітних тренажерах і в системах реального часу (операторські, інженерні, біомедичні інтерфейси, різноманітні системи діагностики і тестування тощо). Саме системні моделі будуть ядром моделювання на сучасному етапі.

Таким чином, модель є абстракцією системи і відображає деякі її властивості. Цілі моделювання формулює дослідник. Значення цілей моделювання неможливо переоцінити. Тільки завдяки їм можна визначити сукупність властивостей модельованої системи, які повинна мати модель, тобто від мети моделювання залежить потрібний ступінь деталізації моделі.

1.3. Співвідношення між моделлю та системою

З огляду на вищеописане модель – це абстракція; вона відображає лише частину властивостей системи, і мета моделювання – визначення рівня абстрактного опису системи, тобто рівня детальності її подання.

Модель і система знаходяться в деяких відношеннях, від яких залежить ступінь відповідності між ними. На міру відповідності між системою та моделлю вказують поняття *ізоморфізму* та *гомоморфізму*. Система та модель є ізоморфними, якщо існує взаємооднозначна відповідність між ними, завдяки якій можна перетворити одне подання на інше. Строго доведений ізоморфізм

для систем різної природи дає можливість переносити знання з однієї галузі в іншу. За допомогою теорії ізоморфізму можна не тільки створювати моделі систем і процесів, але й організувати процес моделювання.

Однак існують і менш тісні зв'язки між системою та моделлю. Це так звані гомоморфні зв'язки, які визначають однозначну відповідність лише в один бік - від моделі до системи. Система та модель є ізоморфними тільки в разі *спрощення* системи, тобто скорочення множини її властивостей (атрибутів) і характеристик поведінки, які впливають на *простір станів системи*. Зазвичай модель простіша за систему.

На рис. 1.1 схематично зображено різницю ізоморфної та гомоморфної залежностей між системою та моделлю для простору станів системи і моделі. Множину станів моделі визначають з огляду на мету моделювання та обраний рівень абстрактного опису.

Отже, *аналогія, абстракція та спрощення* – це основні поняття, які використовуються під час моделювання систем. Розглянемо відношення між системою та моделлю, враховуючи, що ці відношення відповідають цілям моделювання та обмеженням досліджуваної системи.

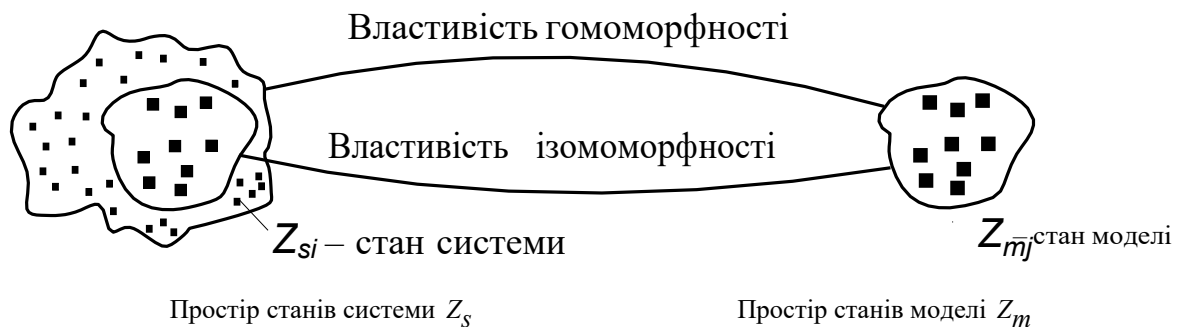


Рис. 1. Схематичне зображення співвідношення між системою і моделлю

1.4. Класифікація моделей

Для того щоб визначити види моделей, перш за все потрібно окреслити ознаки класифікації. У сучасній літературі описано сотні визначень поняття «модель» та їх класифікацій. Одну з перших, досить повних, класифікацій моделей було запропоновано Дж. Форрестером.

Якщо враховувати, що моделювання – це метод пізнання дійсності, то основною ознакою класифікації можна назвати *спосіб подання* моделі. За цією ознакою розрізняють *абстрактні та реальні* моделі (рис.1.2). Під час моделювання можливі різні *абстрактні конструкції*, проте основною є *віртуальна (уявна)* модель, яка відображає ідеальне уявлення людини про навколишній світ, що фіксується в свідомості через думки та образи. Вона може подаватись у вигляді *наочної* моделі за допомогою графічних образів і зображень.

Наочні моделі залежно від способу реалізації можна поділити на дво - або три-мірні графічні, анімаційні та просторові. Графічні та анімаційні моделі широко використовуються для відображення процесів, які відбуваються в модельованій

системі. Графічні моделі застосовуються в системах автоматизованого проектування (CAD). Графічні моделі є базою всіх комп'ютерних ігор, а також застосовуються під час імітаційного моделювання для анімації.

Щоб побудувати модель у *формальному вигляді*, створюють *символічну*, або *лінгвістичну*, модель, яка відповідала б найвищому рівню абстрактного опису, як це було зазначено вище. На базі неї отримують інші рівні опису.

Основним видом абстрактної моделі є математична модель. *Математичною* називається абстрактна модель, яка відображає систему у вигляді математичних відношень.

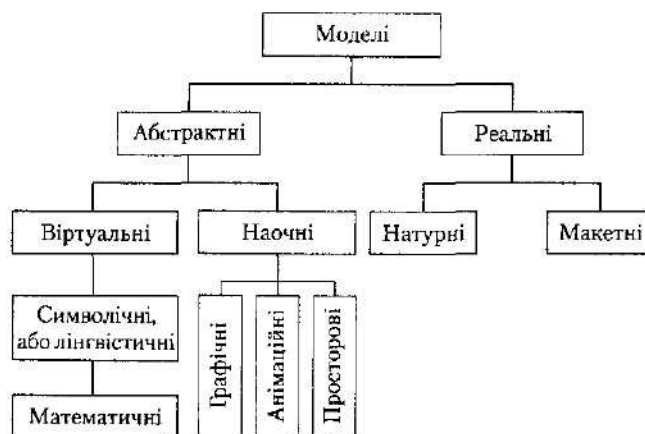


Рисунок 1.2 - Основні типи моделей

Як правило, йдеться про систему математичних співвідношень, що описують процес або явище, яке вивчається; у загальному розумінні така модель є множиною символічних об'єктів і відношень між ними. Математична модель звичайно розглядається як система рівнянь, в якій конкретні величини замінюються математичними поняттями, постійними і змінними величинами, функціями. Як правило, для цього застосовуються диференціальні, інтегральні та алгебричні рівняння. Розвиток нових розділів математики, пов'язаних з аналізом нечислових структур, досвід їх використання під час проведення досліджень свідчать, що потрібно розширювати уявлення про мову математичних моделей. Тоді математична модель визначатиметься як будь-яка математична структура, де об'єкти, а також відношення між ними можна буде інтерпретувати по-різному, наприклад як функції або функціонали.

На відміну від абстрактних, *реальні* моделі існують у природі, й з ними можна експериментувати. Реальні моделі – це такі, в яких хоча б один компонент є фізичною копією реального об'єкта. Залежно від того, в якому співвідношенні знаходяться властивості системи та моделі, реальні моделі можна поділити на натурні та макетні.

Натурні (фізичні) моделі – це існуючі системи або їх частини, на яких провадяться дослідження. Натурні моделі повністю адекватні реальній системі, що дає змогу отримувати високу точність і достовірність результатів моделювання. Суттєві недоліки натурних моделей – це неможливість моделювання критичних і аварійних режимів їх роботи та висока вартість,

Макетні моделі – це реально існуючі моделі, які відтворюють модельовану систему в певному масштабі. Іноді такі моделі називаються масштабними. Параметри моделі та системи відрізняються між собою. Числове значення цієї

різниця називається масштабом моделювання, або коефіцієнтом подібності. Ці моделі розглядаються в рамках теорії подібності, яка в окремих випадках передбачає геометричну подібність оригіналу й моделі для відповідних масштабів параметрів. Найпростіші макетні моделі – це пропорційно зменшені копії існуючих систем, які відтворюють основні властивості системи або об'єкта залежно від мети моделювання. Макетні моделі широко використовуються під час вивчення фізичних та аеродинамічних процесів, гідротехнічних, споруд і багатьох інших технічних систем.

За можливістю змінювати в часі свої властивості моделі поділяються на *статичні* та *динамічні*. Статичні моделі, на відміну від динамічних, не змінюють своїх властивостей у часі. Динамічні моделі також називаються імітаційними.

Залежно від того, яким чином відтворюються в часі стани моделі, розрізняють *дискретні*, *неперервні* й *дискретно-неперервні* (комбіновані) моделі. За відношеннями між станами системи й моделі розрізняють *детерміновані* й *стохастичні* моделі. Останні, на відміну від детермінованих моделей, враховують імовірнісні явища й процеси.

1.5. Вимоги до моделей

У загальному випадку під час побудови моделі потрібно враховувати такі вимоги:

- *незалежність результатів* розв'язання задач від конкретної фізичної інтерпретації елементів моделі;
- *змістовність*, тобто здатність моделі відображати істотні риси і властивості реального процесу, який вивчається і моделюється;
- *дедуктивність*, тобто можливість конструктивного використання моделі для отримання результату;
- *індуктивність* – вивчення причин і наслідків, від окремого до загального, з метою накопичення необхідних знань.

Оскільки модель створюється для вирішення конкретних завдань, розробник моделі має бути впевненим, що не отримає абсурдних результатів, а всі одержані результати відобразатимуть необхідні для дослідника характеристики та властивості модельованої системи. Модель повинна дати можливість знайти відповіді на певні запитання, наприклад: «що буде, якщо ...», оскільки вони є найбільш доцільними під час глибокого вивчення проблеми. Не варто забувати, що системні аналітики використовують модель для прийняття рішень і пошуку найкращих способів створення модельованої системи або її модернізації. Завжди потрібно пам'ятати, що користувачем інформації, отриманої за допомогою моделі, є замовник. Недоцільно розробляти модель, якщо її не можна буде використовувати. Більш того, робота з моделлю має бути автоматизованою для замовника до такого ступеня, щоб він міг працювати з нею в межах своєї предметної області. Таким чином, між моделлю і користувачем повинен бути розвинутий інтерфейс, який звичайно створюється за допомогою системи меню, налагодженої на застосування моделі в певній галузі.

Ступінь деталізації моделі потрібно вибирати з огляду на цілі моделювання, можливості отримання необхідних вхідних даних для моделі та з урахуванням

наявних ресурсів для її створення. Відсутність кваліфікованих фахівців може звести роботи зі створення моделі нанівець, з іншого боку, чим детальніше розроблено модель, тим вона стійкіша до вхідних впливів, які не були передбачені під час проектування, і на більшу кількість запитань може дати правильні відповіді.

1.6. Основні види моделювання

Єдина класифікація видів моделювання неможлива через багатозначність поняття моделі в науці, техніці, суспільстві. Широко відомими видами моделювання є комп'ютерне, математичне, імітаційне та статистичне. На жаль, різні джерела по-різному трактують ці поняття.

Комп'ютерне моделювання визначимо як реалізацію моделі за допомогою комп'ютера. Особливістю комп'ютерного моделювання є його *інтерактивність*, що дає змогу користувачу втручатися в процес моделювання та впливати на результати завдяки узгодженості дій користувача і моделі, яка відтворює об'єкти реального середовища або гіпотетичні події та процеси. Під час комп'ютерного моделювання можуть бути задіяні реальні об'єкти, віртуальні об'єкти, згенеровані комп'ютером, які відтворюють реальні об'єкти (наприклад, потоково-конвеєрна лінія для збирання автомобілів). Інтерактивне комп'ютерне моделювання широко застосовується в навчальних системах, наприклад для побудови тренажерів і в ситуаційних іграх.

Термін «моделювання» відповідає англійському слову «modeling», тобто побудові моделі та її аналізу. Англійський термін «simulation» відповідає прийнятому терміну «імітаційне моделювання», але часто вони використовуються разом, коли йдеться про технологічні або системні етапи моделювання, пов'язані з прийняттям рішень за допомогою моделей.

Імітаційне моделювання – це метод конструювання моделі системи та проведення експериментів. Однак під таке визначення підпадають майже всі види моделювання. Тому потрібно виділити суттєві особливості імітаційного моделювання.

Перш за все слід подати в моделі *структуру* системи, тобто загальний опис елементів і зв'язків між ними, потім визначити засоби відтворення в моделі поведінки системи. Здебільшого поведінку системи описують за допомогою станів і моментів переходів між ними. Стан системи в момент часу i визначають як безліч значень певних параметрів системи у цей самий момент часу i . Будь-яку зміну цих значень можна розглядати як перехід до іншого стану. І врешті-решт, імітаційна модель має відобразити властивості середовища, в якому функціонує досліджувана система. Зовнішнє середовище задають вхідними впливами на модель.

Вся інформація про імітаційну модель загалом має логіко-математичний характер і подається у вигляді сукупності алгоритмів, які описують процес функціонування системи. Отже, здебільшого імітаційною моделлю є її програмна реалізація на комп'ютері, а імітаційне моделювання зводиться до проведення експериментів з моделлю шляхом багаторазового прогону програми з деякою множиною даних – середовищем системи. Під час

імітаційного моделювання можуть бути задіяні не тільки програмні засоби, але й технічні засоби, люди та реальні системи.

Із математичної точки зору імітаційну модель можна розглядати як сукупність рівнянь, які розв'язують з використанням чисельних методів у разі кожної зміни модельного часу. Окремі рівняння можуть бути простими, але їх кількість і частота розв'язання \sim дуже великими. Розв'язання таких рівнянь під час імітаційного моделювання означає *встановлення хронологічної послідовності подій*, які виникають у системі і відображають послідовність її станів. Отже, імітаційна модель функціонує так само, як система.

моделювання, що виконується за допомогою методу *статистичного моделювання* (методу Монте-Карло). Статистичне моделювання є самостійним видом моделювання, яке включається в імітаційне моделювання тільки за необхідності моделювання ймовірнісних систем і процесів.

Із вищенаведеного прикладу видно, що стохастичне моделювання використовується під час імітаційного моделювання тільки за необхідності врахування випадкових факторів.

1.7. Формальні методи побудови моделей

Розглядаючи сфери застосування моделей, можна констатувати, що за допомогою моделі можна досягти *двох основних цілей*: *описової*, якщо модель призначена для пояснення і кращого розуміння об'єкта, або *приписуючої*, коли модель дає змогу передбачити або відтворити характеристики об'єкта або визначити його поведінку. Таким чином, *модель є описовою*, якщо вона призначена зображувати поведінку (функціонування) або властивості існуючої або типової системи (наприклад, масштабна модель або письмовий опис, що дає змогу знайомити потенційних покупців із фізичними і робочими характеристиками комп'ютера). Протилежність - *приписуюча модель*, яка відображає необхідну поведінку або властивості запропонованої системи (наприклад, масштабна модель або письмовий опис, представлений постачальнику комп'ютерів, з фізичними і робочими характеристиками потрібного замовнику комп'ютера).

Приписуюча модель може бути описовою, але не навпаки. Тому існує різний ступінь корисності моделей, які використовуються в технічних і соціальних науках. Це значною мірою залежить від методів, і засобів, застосовуваних під час побудови моделей, а також від кінцевої мети. У соціальних науках моделі призначено для пояснення існуючих систем, а в техніці вони є допоміжними засобами для створення нових або більш досконалих моделей. Модель, що придатна для досягнення цілей розроблення системи, має також пояснювати її.

Під час побудови моделей застосовуються фундаментальні закони природи, варіаційні принципи, аналогії, ієрархічні ланцюжки. Процес створення моделі включає такі етапи.

1. Словесно-смысловий опис об'єкта або явища – формулювання *описової* моделі, призначеної для сприяння кращому розумінню об'єкта моделювання.
2. Числове вираження модельованої реальності для виявлення кількісної міри і границь відповідних якостей; з цією метою провадиться математико-

статистична обробка емпіричних даних, пропонується кількісне формулювання якісно встановлених фактів і узагальнень.

3. Перехід до вибору або формулювання моделей явищ і процесів (варіаційного принципу, аналогії тощо) і його запису у формалізованій формі; це рівень структурних теоретичних схем, таких як системи масового обслуговування, мережі Петрі, скінченні або ймовірнісні автомати, діаграми фонд–потік тощо.

4. Завершення формулювання моделі її „оснащенням” – задання початкового стану і параметрів об'єкта.

5. Вивчення моделі за допомогою доступних методів (у тому числі із застосуванням різних підходів і обчислювальних методів).

У результаті дослідження моделі досягається поставлена мета. У цьому разі має бути встановлена всіма можливими способами (шляхом порівняння з практикою, порівняння з іншими підходами) її адекватність – відповідність об'єкта сформульованим припущенням.

1.8. Кібернетичний підхід

Систему можна вивчати та аналізувати, змінюючи вхідні впливи і спостерігаючи за виходами. Це кібернетичний підхід, згідно з яким система розглядається як «чорний ящик». Метод „чорного ящика”- широко використовується під час моделювання систем, коли для дослідника важливо отримати інформацію про поведінку системи, а не про її будову. Дослідник не може зробити однозначний висновок про структуру «чорного ящика», спостерігаючи тільки за його входами та виходами, бо поведінка модельованої системи нічим не відрізняється від поведінки ізоморфних їй систем.

Для побудови моделі використовуються методи теорії ідентифікації. У загальному випадку завдання ідентифікації формулюється так: на основі результатів спостереження за вхідними та вихідними змінними системи потрібно побудувати оптимальну в деякому розумінні математичну модель. Основними етапами ідентифікації є такі:

1. Вибір класу і структури моделі та мови її опису.
2. Вибір класу і типів вхідних впливів X .
3. Обґрунтування критеріїв подібності системи та моделі.
4. Вибір методу ідентифікації та розроблення відповідних алгоритмів оцінювання параметрів системи.
5. Перевірку адекватності отриманої в результаті ідентифікації моделі.

Залежно від обсягу апріорної інформації про клас і структуру системи вирізняють завдання ідентифікації в широкому та вузькому розумінні. Завдання ідентифікації у широкому розумінні виконується в умовах апріорної невизначеності структури моделі системи («чорний ящик»). Клас і структура математичної моделі вибираються на основі результатів теоретичного аналізу з використанням загальних закономірностей процесів, які протікають у системі, або на основі загальної інформації про подібні системи. У цьому випадку для побудови математичної моделі можна використовувати непараметричні методи. Їх розроблено для тих ситуацій, які досить часто виникають на практиці, коли

дослідник нічого не знає про параметри досліджуваної системи (звідси і назва методів – *непараметричні*).

Завдання ідентифікації у вузькому розумінні полягає в оцінюванні параметрів і станів системи, якщо відома структура моделі («сірий ящик»). Завданням ідентифікації є кількісне оцінювання певних параметрів. Для цього використовується параметрична ідентифікація математичної моделі. Прикладами таких моделей можуть бути диференціальні та різницеві рівняння, моделі типу «вхід-стан-вихід».

На рис. 1.3 зображено загальну схему ідентифікації системи. Вхідні впливи X на систему та модель однакові, виходи системи Y_s і моделі Y_m у загальному випадку відрізняються. Для їх порівняння потрібно сформулювати критерій подібності та мінімізувати його, тобто налагодити модель.

Прикладами моделей, створених на основі експериментальних даних, можуть бути моделі авторегресії різних порядків, ковзного середнього та моделі типу «вхід-вихід», побудовані за допомогою методу найменших квадратів.

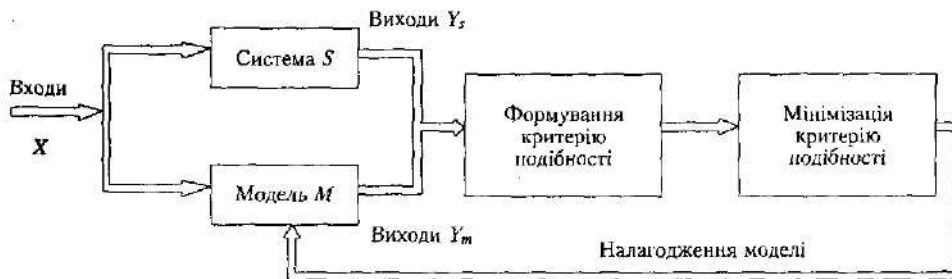


Рисунок 1.3 – Загальна схема ідентифікації системи

1.9. Теоретико-множинний підхід

Згідно з теоретико-множинним підходом формальна модель динамічної системи має такий вигляд:

$$M = \langle T, X, Y, Z, z(t), P \rangle,$$

де T – модельний час; X, Y – множина відповідно вхідних і вихідних змінних; Z – простір станів моделі; $z(t)$ – функція станів, $i \in T$; P – множина процесів, яка визначається як множина впорядкованих у часі пар елементів $\langle x, z(\tau, t) \rangle$, де $t \in$

$a \tau$ – початковий момент модельного часу для процесу $p \in P$. Таке визначення задає модель системи у вигляді *схеми процесів*, у якій множини процесів можуть існувати паралельно в модельному часі T .

Вважається, що деяка подія з множини подій C зумовлює зміну стану системи, якщо починається певний процес або закінчується деякий процес $p_j \in$

У протилежному випадку стан системи $p_i \in P$ не змінюється. Тоді можна задати *подійну схему* моделі:

$$M = \langle T, X, Y, Z, z(t), C \rangle,$$

де $C \sim$ множина подій, що визначається як множина впорядкованих у часі елементів $\langle j, d[\tau, t_j] \rangle$, де $c \in C$, $d[\tau, t_j]$ – функція дії для процесу $p_i \in P$; $t \in T$, а

початковий момент модельного часу T . У цій схемі процес моделювання записується як послідовність подій, що відбуваються в моделі.

Припустимо, що завдяки виконанню деякої умови u з множини U почне виконуватись певна дія $d[\tau, t_j]$ з множини для деякого процесу $p_j \in P$. Тоді можна задати модель системи у вигляді *схеми дій*: $M = \langle T, X, Y, Z, z(t), D \rangle$.

У цій схемі процес моделювання описується як перевірка всіх умов у разі кожної зміни модельного часу $t \in T$, щоб знайти умову, яка почне певну дію з множини D . Зміна часу t може відбуватись з постійним або змінним від події до : кроком. Схеми моделей широко застосовуються під час побудови алгоритмів моделювання і мов дискретного імітаційного моделювання.

Якщо припустити, що виконання деякої множини процесів P може призвести до зміни станів $z \in Z$ і виникнення нових процесів, що спричинить появу деякої множини ситуацій L , тобто $z(t) : Pz \rightarrow L$, то отримуємо ситуаційну або причинно-наслідкову схему:

$$M = \langle T, X, Y, Z, z(t), L \rangle,$$

в якій потрібно описати множину ситуацій та множину правил (алгоритми якими визначають процес, що має виконуватись. Поведінка моделі в таких с мах зображується у вигляді ланцюга

$$\{\text{ситуація}\} \rightarrow \{\text{правило}\} \rightarrow \{\text{процес}\}.$$

Якщо модель здатна конструювати нові правила на основі існуючих, то перетворюється на модель зі штучним інтелектом.

Під час ситуаційного моделювання, як правило, повний опис усіх можливих ситуацій замінюється деякою множиною узагальнених ситуацій, кожна з певною мірою ймовірності відтворює один із можливих станів системі кожної ситуації існує набір правил дії. Вибір того або іншого правила може здійснюватись за деяким критерієм або за допомогою таблиць прийняття рішень, а у простіших випадках - згідно із заданою ймовірністю. Моделювання виконується шляхом програвання різних ситуацій за певним сценарієм, яким у окремому випадку

може бути алгоритм моделювання. Таким чином створюють різні ігри, наприклад ділові, військові, економічні, розважальні. *Гра* – це спрощене відтворення реального процесу, яке здебільшого використовується для навчання, прийняття рішень, проведення досліджень або розваг.

Визначити систему можна не тільки як сукупність елементів, але і як сукупність відношень, спостерігаючи за їх змінами. Перш за все це стосується взаємодії між різними динамічними системами, кожна з яких досить складна. Прикладом можуть бути екологічні та соціальні системи. Під час вивчення таких систем дослідник, базуючись на системному аналізі, вивчає та описує впливи однієї системи на іншу.

1.10. Принципи побудови моделей

Розглянемо коротко основні принципи моделювання, які відображають достатньо багатий досвід, накопичений на даний час у галузі розроблення і використання моделей.

◆ **Принцип інформаційної достатності.** За повної відсутності інформації про систему модель побудувати неможливо. За наявності повної інформації про систему її моделювання недоцільне. Існує деякий критичний рівень апріорних

відомостей про систему (рівень інформаційної достатності), після досягнення якого можна побудувати її адекватну модель.

◆ **Принцип доцільності.** Модель створюється для досягнення деяких цілей, які визначають на первинному етапі формулювання проблеми моделювання.

◆ **Принцип здійсненності.** Модель, яка створюється, має забезпечувати досягнення мети дослідження з урахуванням граничних ресурсів з імовірністю, суттєво відмінною від нуля, і за кінцевий час. Звичайно задають деяке граничне значення P (ступінь ризику) ймовірності досягнення мети моделювання $P(\xi)$, а також сам граничний термін i досягнення мети. Модель вважають здійсненою, якщо $P(i) > P$.

◆ **Принцип множинності моделей.** Модель, яка створюється, має відображати в першу чергу ті властивості реальної системи (або явища), які впливають на вибраний показник ефективності. Відповідно під час використання будь-якої конкретної моделі пізнаються лише деякі складові реальності. Для повного її дослідження необхідно мати ряд моделей, які дали б змогу відобразити певний процес з різних боків і з різним ступенем детальності.

◆ **Принцип агрегації.** У більшості випадків складну систему можна подати такою, що складається з агрегатів (підсистем), для адекватного формального описування яких придатними є деякі стандартні математичні схеми. Принцип агрегації дає змогу досить гнучко перебудувати модель залежно від завдань дослідження.

◆ **Принцип параметризації.** У ряді випадків модельована система має у своєму складі деякі відносно ізольовані підсистеми, які характеризуються певними параметрами, у тому числі векторними. Такі підсистеми можна замінювати

в моделі відповідними числовими величинами, а не описувати процес їх функціонування. У разі необхідності залежність значень цих величин від ситуації може задаватись у вигляді таблиць, графіків або аналітичних виразів (формул), наприклад за допомогою регресійного аналізу. Принцип параметризації дає змогу скоротити обсяг і тривалість моделювання, але слід мати на увазі, що параметризація знижує адекватність моделі.

1.11. Технологія моделювання: основні етапи, їх взаємозв'язок та характеристики

Основою моделювання є методологія системного аналізу. Це дає можливість досліджувати систему, яка проектується або аналізується, за технологією операційного дослідження. Комп'ютерне моделювання (а зазвичай застосовується саме комп'ютерна модель) включає такі взаємопов'язані етапи :

1. Формулювання проблеми і змістовна постановка задачі.
2. Розробка концептуальної моделі.
3. Розробка програмної реалізації моделі, яка включає:
 - а) вибір засобів програмування, за допомогою яких буде реалізована модель;
 - б) розробка структурної схеми моделі і складання опису її функціонування;

- в) програмна реалізація моделі.
- 4. Перевірка адекватності моделі.
- 5. Організація і планування проведення експериментів, що включає оцінювання точності результатів моделювання.
- 6. Інтерпретація результатів моделювання і прийняття рішень.
- 7. Оформлення результатів дослідження.

На першому етапі замовник формулює проблему. Організуються зустрічі керівника проекту із замовником, аналітиками з моделювання і експертами із проблеми, що вивчається. Визначаються цілі дослідження і спеціальні питання, відповіді на які будуть отримані в результаті дослідження; встановлюються критерії оцінювання роботи, які використовуватимуться для вивчення ефективності різних конфігурацій системи. Розглядаються такі показники, як масштаб моделі, період дослідження і необхідні ресурси; визначаються конфігурації модельованої системи, а також необхідне програмне забезпечення.

На цьому ж етапі ведеться цілеспрямоване дослідження модельованої системи, притягуються експерти з вирішуваної проблеми, які володіють достовірною інформацією. Збирається інформація про конфігурацію системи і способи її експлуатації для визначення параметрів моделі і вхідних розподілів ймовірностей.

На другому етапі розробляється **концептуальна модель** — абстрактна модель, яка дає можливість виявити причинно-наслідкові зв'язки, властиві досліджуваному об'єкту в межах, визначених цілями дослідження. По суті, це формальний опис об'єкта моделювання, яке відображає концепцію (погляд дослідника на проблему). Вона включає в явному вигляді логіку, алгоритми, припущення й обмеження.

Згідно з цілями моделювання визначаються вихідні показники, які потрібно збирати під час моделювання, ступінь деталізації, необхідні початкові дані для моделювання.

Рівень деталізації моделі залежить від таких факторів: мети проекту; критеріїв оцінювання показників роботи; доступності даних; достовірності результатів; комп'ютерних можливостей; думки експертів з вирішуваної проблеми; обмежень, пов'язаних з часом і фінансуванням. Ведеться структурний аналіз концептуальної моделі, пропонується опис допущень, які обговорюються з замовником, керівником проекту, аналітиками й експертами з вирішуваної проблеми.

Розробляються моделі початкових даних, ведеться їх статистичний аналіз, за результатами якого визначають розподіли ймовірності, регресійні, кореляційні й інші залежності. На цьому етапі для попереднього аналізу даних широко застосовують різні статистичні пакети (наприклад, STATISTICA).

Для динамічних систем ведеться післяопераційний аналіз функціонування модельованої системи з детальним описом роботи елементів системи. За результатами такого аналізу можна з'ясувати, чи можна вирішити проблему без застосування засобів моделювання. Детально оброблена концептуальна модель дає можливість замовникові з іншого боку поглянути на роботу

системи і, наприклад, визначити її вузькі місця, які сприяють зниженню її пропускну́ї спроможності.

Одна зі складних проблем, з якою має справу аналітик моделювання, полягає у визначенні, адекватності моделі системі. Якщо імітаційна модель "адекватна", її можна використовувати для прийняття рішень щодо системи, яку вона представляє, тобто нібито вони приймалися на основі результатів проведення експериментів з реальною системою. Модель складної системи може тільки приблизно відповідати оригіналу, незалежно від того, скільки зусиль було витрачено на її розробку, оскільки абсолютно адекватних моделей не існує.

Оскільки модель завжди повинна розроблятися для певного набору цілей, то модель, яка є адекватною для однієї мети, може не бути такою для дослідження іншої мети. Слід зазначити, що адекватна модель не обов'язково є достовірною, і навпаки. Модель може бути достовірною, але, в цьому випадку, не використовуватися для прийняття рішень. Наприклад, достовірна модель може не бути адекватною з політичних або економічних причин.

При програмній реалізації моделі визначаються засоби програмування, тобто мови програмування або пакети. Наприклад, можуть використовуватися мови програмування загального призначення, такі, як C+ чи спеціалізовані засоби для моделювання (наприклад, Agena, Automod, Extend, GPSS, iThink). Перевага використання мов програмування полягає в тому, що, як відомо, вони мають невисоку закупівельну вартість і на розробку моделі з їх допомогою витрачається менше часу. Разом з тим використання спеціалізованого програмного забезпечення для моделювання призводить до зменшення тривалості процесу програмування і вартості всього проекту.

Серед спеціалізованих пакетів для моделювання слід зазначити пакет MATLAB з інтерактивним модулем Simulink. Пакет MATLAB є всесвітньо визнаним універсальним відкритим середовищем і мовою програмування одночасно, в якому інтегровані засоби обчислень, візуалізації, програмування і моделювання.

Здійснюється програмування моделі та її відлагоджування, виконуються тестові прогони моделі на основі контрольних даних, ведеться аналіз чутливості, щоб визначити, які фактори в моделі суттєво впливають на робочі характеристики системи і повинні моделюватися точніше.

Після кожного з вищезазначених етапів перевіряється достовірність моделі. Перевірку умовно можна розділити на два етапи: перевірка правильності створення концептуальної моделі, тобто задуму — *валідація*; перевірка правильності її реалізації — *верифікація*. Під час перевірки достовірності потрібно відповісти на питання про відповідність моделі модельованій системі, тобто визначити, наскільки ізоморфні система і модель. Як правило, в разі моделювання вимога ізоморфізму об'єкта і моделі надмірна, оскільки в цьому випадку складність моделі повинна відповідати складності об'єкта. Тому будують гомоморфні моделі, в яких виконується вимога однозначної відповідності моделі об'єкту.

На етапі *верифікації* розглядають, чи правильно реалізована концептуальна модель (модельні припущення) в комп'ютерну програму, тобто виконують налагоджування програми моделювання. Це складне завдання, оскільки може існувати багато логічних шляхів.

Етап перевірки правильності реалізації моделі включає перевірку еквівалентності перетворення моделі на кожному з етапів її реалізації і порівняння станів. У цьому випадку модель зазнає таких змін: концептуальна модель — математична модель — алгоритм моделювання — програмна реалізація моделі.

Валідація – це процес, що дає можливість встановити, чи є модель (а не комп'ютерна програма) достатньо точним відображенням системи **для конкретних цілей дослідження**.

Розробляється план проведення експериментів з моделлю для досягнення поставлених цілей. Основна мета планування експериментів - вивчення поведінки модельованої системи при найменших витратах під час експериментів. Зазвичай проводять такі експерименти:

- порівнюють середні значення і дисперсії різних альтернатив;
- визначають важливість врахування впливу змінних і обмежень, які накладаються на ці змінні;
- визначають оптимальні значення з деякої множини можливих значень змінних.

Проведення експериментів планують для пошуку незначущих факторів. У разі оптимізації якого-небудь числового критерію формулюють гіпотези щодо вибору якнайкращих варіантів структур модельованої системи або режимів її функціонування, визначають діапазон значень параметрів (режимів функціонування) моделі, в межах якого знаходиться оптимальне рішення. Визначають кількість реалізацій і час прогону моделі кожної реалізації. Проводять екстремальний експеримент, за результатами якого знаходять оптимальне значення критерію і відповідні значення параметрів. Для оцінювання точності стохастичних моделей будують довірчі інтервали для отримуваних вихідних змінних.

Далі аналізують і оцінюють результати. Представляють результати комп'ютерних експериментів у вигляді графіків, таблиць, роздруківок, а також визначають якісні і кількісні оцінки результатів моделювання. Для візуалізації моделі використовують анімацію. Обговорюють процес створення моделі і її достовірність, щоб підвищити рівень довіри до неї.

За отриманими результатами формулюють висновки з проведених досліджень і визначають рекомендації щодо використання моделі прийняття рішень.

Вищенаведені етапи моделювання взаємозв'язані, а сама процедура створення моделі є ітераційною. Це пояснюється тим, що після виконання кожного етапу перевіряється правильність і достовірність моделі і в разі невідповідності моделі об'єкту здійснюється повернення до попередніх етапів з метою корегування і настройки моделі. Залежно від характеру внесених змін повертаються безпосередньо до попереднього етапу або до попередніх етапів.

На останньому етапі моделювання документально оформляють всі результати дослідження і готують програмну документацію для використання їх під час розробки поточних і майбутніх проєктів.

Висновки

- ◆ Система – це цілісний комплекс взаємопов'язаних елементів, який має певну структуру і взаємодіє із зовнішнім середовищем.
- ◆ Модель – це реально існуюча або уявна система, яка, заміщаючи і відображаючи в пізнавальних процесах іншу систему-оригінал, знаходиться з нею у відношенні подібності.
- ◆ Моделювання – це спосіб дослідження будь-яких явищ, процесів або об'єктів шляхом побудови та аналізу їх моделей.
- ◆ Імітаційне моделювання -- це метод конструювання моделі системи та проведення експериментів над моделлю.

Контрольні запитання та завдання

1. Що таке система? Як впливає на систему зовнішнє середовище? Чому існує багато визначень системи?
2. Назвіть кілька статичних і динамічних об'єктів, дій, процесів, атрибутів, подій та змінних станів для таких систем:
 - а) станція технічного обслуговування автомобілів;
 - б) магазин самообслуговування;
 - в) станція швидкої допомоги;
 - г) кафе;
 - д) таксомоторний парк.
3. Яким чином динамічна поведінка системи пов'язана з поняттям стану системи?
4. Що розуміють під абстрактною системою?
5. Що розуміють під моделлю? У яких відношеннях перебувають об'єкт моделювання та модель? Чи може система бути моделлю?
6. Яку роль відіграють поняття стану та процесу в моделюванні? Опишіть роботу банку з двома касирами, до яких стоїть одна черга. Виділіть основні стани цієї системи.
7. Наведіть приклади задач, які можна розв'язати за допомогою моделювання. В яких випадках задачі можна розв'язати лише у такий спосіб?
8. Виконайте критичний аналіз різних видів класифікацій моделей та видів моделювання. Чому неможлива єдина класифікація? Запропонуйте іншу класифікацію моделей.
9. Покажіть, яким чином можна провести декомпозицію для нескінченних множин.
10. Сформулюйте завдання ідентифікації в широкому та вузькому розумінні для задачі про водопостачання.
11. Яким чином задається час моделювання в задачах про водопостачання? Чи можливо так задати час моделювання для цієї задачі, щоб він залежав від деяких подій? Наведіть приклади моделювання таких подій.
12. Дайте ситуаційний опис переходу пішоходом дороги. Розгляньте всі можливі ситуації.

ТЕМА 2

МОДЕЛІ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ

2.1. Характеристики систем масового обслуговування

2.1.1. Вхідний потік вимог

2.1.2. Організація черги

2.2. Режими роботи системи масового обслуговування

2.3. Типи моделей систем масового обслуговування

2.4. Основи дискретно-подійного моделювання систем масового обслуговування

2.5. Загальні відомості про мережі СМО

Висновки

Контрольні запитання

У теорії і практиці моделювання систем важливе місце посідають моделі систем масового обслуговування (СМО). Такі системи зустрічаються нам щоденно. Це процеси обслуговування в черзі на заправній станції, у магазині, бібліотеці, кафе, також різні служби ремонту і медичної допомоги, транспортні системи, аеропорти, вокзали тощо. Черги виникають і за потреби скористатись телефонним зв'язком або передати повідомлення по Інтернету. Більше того, будь-яке виробництво також можна подати як послідовність таких систем. Особливого значення СМО набули в інформатиці, Це передусім комп'ютерні системи, мережі передавання інформації, бази і банки даних.

Існує розвинутий математичний апарат теорії масового обслуговування (науковці західних країн цю теорією називають теорією черг), що дає змогу аналізувати ефективність функціонування СМО певних типів і визначати залежність між характеристиками потоку вимог, кількістю каналів (пристроїв для обслуговування), їх продуктивністю, правилами роботи СМО та її ефективністю.

Перші теоретичні результати внаслідок вирішення проблем, пов'язаних із функціонуванням систем обслуговування, було отримано датським ученим, співробітником Копенгагенської телефонної компанії А. К. Ерлангом. Ці результати стосувались практичних завдань підвищення якості обслуговування абонентів і визначення кількості телефонних ліній. У подальшому з'ясувалось, що отримані теоретичні результати є настільки загальними, що їх можна використовувати для визначення оптимальної кількості кас і продавців на торговельних підприємствах, для розрахунків запасів у магазинах, достатніх для їх безперебійної роботи, тощо. Однак більшість результатів було отримано для систем, в яких процеси надходження та обслуговування вимог є марківськими або напівмарківськими. У цьому разі завдання аналізу СМО можна описати звичайними диференціальними рівняннями і в явному вигляді обчислити основні характеристики системи.

На практиці часто виникають задачі, пов'язані з чергами, які неможливо розв'язати із застосуванням існуючих методів теорії масового обслуговування. Це зумовило інтенсивний розвиток методів дослідження СМО за допомогою засобів імітаційного моделювання. У цьому випадку характеристики СМО оцінюються наближено шляхом обробки результатів моделювання системи.

У даному розділі розглядаються обидва підходи до аналізу СМО, причому основна увага зосереджується на методах імітаційного моделювання. Зазначається, яким чином потрібно будувати алгоритми моделювання та їх програмні реалізації для дискретно-подійних систем і як збирати інформацію про показники їх роботи.

2.1. Характеристики систем масового обслуговування

Функціонування будь-якої СМО полягає в обслуговуванні потоку вимог, які одна за одною або групами надходять до неї в деякі, як правило, випадкові моменти часу. Вимоги, які надійшли до СМО, обробляються протягом певного часу, після чого залишають систему.

У будь-якій системі обслуговування передбачена наявність *пристроїв для обслуговування* (інші назви: *прилади для обслуговування, сервери, канали*) і *вимог* (інші назви: *заявки, виклики, клієнти*), які потребують обслуговування. Правила або алгоритми взаємодії пристроїв і вимог називатимемо *дисциплінами поставлення в чергу та обслуговуванням*.

Для кожної СМО задається режим роботи. Слід відзначити, що для вимоги може бути потрібно кілька обслуговувань одним або кількома пристроями. Звичайно термін *пристрій для обслуговуванням* (англійською – «SERVER») використовується для відносно простих моделей, в яких кожна вимога може обслуговуватись тільки одним пристроєм. Якщо ж вимоги обслуговуються кількома пристроями в певній послідовності, переміщаючись за заданим маршрутом, то має місце «мережа обслуговування» (англійською – «queueingnetwork»). Іншими словами, мережа – це складна СМО.

Зазвичай за допомогою методів теорії масового обслуговування розв'язують задачі з проектування та експлуатації однотипних елементів обслуговування – наприклад, розраховують кількість контрольно-пропускного обладнання, місць для ремонту, бензоколонок, обслуговуючого персоналу, ліній зв'язку, одиниць обладнання обчислювальної техніки тощо.

Окремим типом завдань у теорії масового обслуговування є визначення місць накопичування вимог у системі обслуговування, наприклад визначення місць на стелажах на складі або в багатопверховому гаражі, кількості пристроїв введення-виведення інформації комп'ютера, кількості місць у палатах госпіталю та ін.

Найчастіше ефективність функціонування будь-якої СМО визначається за такими показниками:

- ◆ середня кількість вимог, які система може обслужити за одиницю часу;
- ◆ середній відсоток вимог, які не були обслужені;
- ◆ ймовірність того, що вимогу, яка надійшла до системи, буде прийнято для обслуговування;
- ◆ середній час очікування вимоги в черзі; 4 закон розподілу часу очікування;
- ◆ середня кількість вимог у черзі;
- ◆ закон розподілу числа вимог у черзі;
- ◆ коефіцієнт завантаження пристрою для обслуговування; 4 середня кількість пристроїв, зайнятих обслуговуванням.

Щоб визначити ці параметри, потрібно охарактеризувати СМО, тобто описати та задати такі характеристики:

- ◆ вхідний потік вимог (вимоги, які надходять до системи для обслуговування);
- ◆ дисципліни постановки вимог у чергу та вибору вимог із неї;
- ◆ правила, за якими здійснюється обслуговування;
- ◆ вихідний потік вимог (вимоги, які залишають систему);
- ◆ режими роботи системи.

Розглянемо ці характеристики більш детально.

2.1.1. Вхідний потік вимог

Для визначення вхідного потоку вимог потрібно зазначити моменти часу їх надходження до системи (закон надходження) і кількість вимог, які можуть надійти одночасно. Закон надходження може бути детермінованим (наприклад, вимога або вимоги надходять до системи у фіксовані моменти часу) або ймовірнісним (проміжки часу між моментами надходження вимог до системи мають рівномірний, експоненціальний або інший заданий закон розподілу).

У загальному випадку вхідний потік вимог описується розподілом імовірностей проміжків часу між моментами надходження до системи двох сусідніх вимог. Здебільшого припускається, що ці проміжки часу є незалежними і мають однаковий розподіл випадкових величин i , таким чином, вимоги утворюють стаціонарний вхідний потік. У класичній теорії масового обслуговування, як правило, розглядається так званий *пуассонівський* (найпростіший) потік вимог, в якому кількість вимог k для будь-якого проміжку часу t має розподіл Пуассона: де λ – інтенсивність потоку вимог (кількість вимог, які надійшли до системи за одиницю часу).

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad k \geq 0, \quad t \geq 0,$$

Існує кілька теорем теорії ймовірностей, за допомогою яких можна довести, що більшість потоків вимог є пуассонівськими. Це, зокрема, теореми про суперпозицію незалежних потоків малої інтенсивності, розрідження випадкового потоку та ін. На практиці вважають, що вхідний потік має розподіл Пуассона, якщо за визначений проміжок часу вимоги надходять до системи під великої кількості незалежних джерел. Прикладами таких потоків можуть бути дзвінки абонентів у телефонній мережі або запити до централізованої бази даних від користувачів комп'ютерної мережі.

Для задання вхідного потоку вимог, крім закону розподілу, потрібно визначити кількість вимог, які надходять до системи одночасно. Вимоги можуть надходити до системи по одній або групами, наприклад у метро або до стадіону через вхідні турнікети одночасно можуть зайти кілька чоловік. Системи, до яких вимоги надходять пакетами (більше ніж з однією вимогою), будемо називати системами з груповим потоком вимог.

Кількість вимог, які надходять до системи від якогось джерела (під час моделювання це джерело відтворює генератор вимог), може бути необмеженою або обмеженою. Прикладом системи з обмеженою кількістю вимог є система з відмовами та відновленням (ремонт) обладнання виробничої дільниці, де в разі відмови обладнання подається вимога на ремонт обладнання до бригади

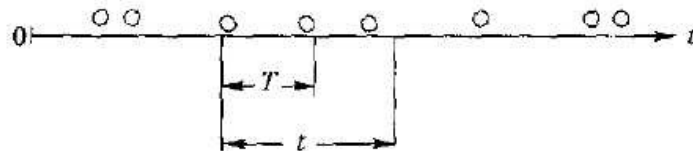
ремонтників. Якщо в такій системі є M одиниць обладнання, то максимально можлива кількість вимог у системі дорівнюватиме M .

Прикладом системи з необмеженою кількістю вимог може бути телефонна мережа, кількість абонентів якої визначити практично неможливо. Якщо кількість вимог безмежна і вони незалежні, то це свідчить про те, що вхідний потік вимог є пуассонівським.

Моделювання пуассонівського потоку

Покажемо, як під час моделювання систем масового обслуговування можна задати пуассонівський потік вимог. Для цього розглянемо найпростіший потік з інтенсивністю λ і позначимо моменти надходження вимог на осі $(0, t)$, як показано на рис. 2.1. Визначимо, який розподіл мають проміжки часу між моментами надходження двох сусідніх вимог.

Очевидно, що проміжки часу T - випадкові величини. Знайдемо закон їх розподілу. Функція розподілу $F(t)$ визначає ймовірність того, що випадкова величина T набуде значення, яке менше за t , тобто



$$F(t) = P, \quad T < t.$$

Рисунок 2.1 – Моменти надходження вимог для пуассонівського потоку

Нехай t_0 – початок проміжку часу t . Знайдемо ймовірність того, що випадкова величина T буде меншою за t . Для цього потрібно, щоб на проміжок довжиною t , який починається з точки t_0 , потрапила хоча б одна вимога. Обчислимо функцію $F(t)$ через ймовірність протилежної події, тобто через ймовірність P_0 того, що за проміжок часу t до системи не надійде жодної вимоги:

Значення ймовірності P_0 знайдемо за формулою (2.1) за умови, що $\lambda t = 0$:

- Тоді функція розподілу випадкової величини T матиме вигляд:

$$F(t) = 1 - e^{-\lambda t}, \quad t > 0.$$

$$F(t) = 1 - P_0.$$

$$P_0 = \frac{(\lambda t)^0}{0!} e^{-\lambda t} = e^{-\lambda t}.$$

$$f(t) = \lambda e^{-\lambda t}, \quad t > 0.$$

Це і є функція щільності показникового або експоненціального закону розподілу, її графік і графік функції розподілу зображено на рис. 2.2.

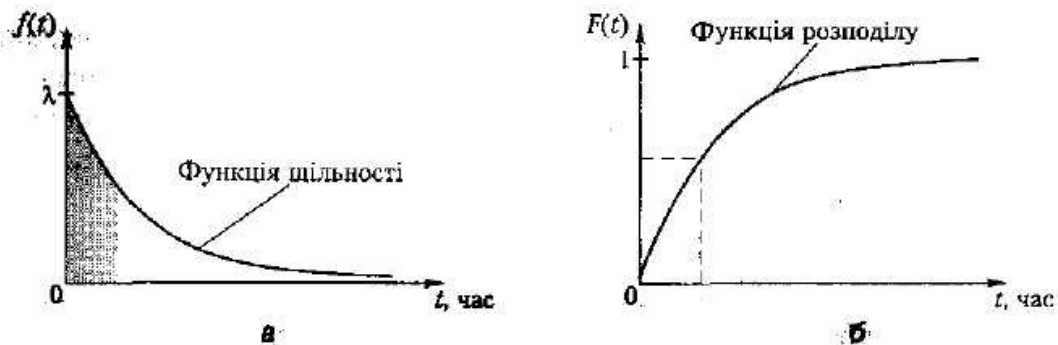


Рисунок 2.2 - Графіки функцій щільності та розподілу експоненціального закону

Щоб знайти функцію щільності розподілу $f(t)$ випадкової величини T , продиференціюємо функцію $F(t)$ за t :

Отже, щоб отримати пуассонівський потік вхідних вимог, які надходять до системи, достатньо обчислити випадкову величину з експоненціальним розподілом.

Властивості пуассонівського потоку

У теорії масового обслуговування найпростіший потік відіграє таку ж роль, як нормальний закон розподілу випадкових величин у теорії ймовірностей. Випадковий потік вимог, який за своїми характеристиками наближений до найпростішого, утворюється в разі додавання випадкових потоків. Основними властивостями найпростішого потоку вимог є:

- ◆ стаціонарність;
- ◆ відсутність післядії;
- ◆ ординарність.

Потік є *стаціонарним*, якщо ймовірність надходження певної кількості вимог за деякий проміжок часу t залежить тільки від довжини цього проміжку t і параметра X , і не залежить від місця розташування проміжку на осі відрізки часу не повинні перетинатися.

У потоці *відсутня післядія*, якщо ймовірність надходження визначеної кількості вимог за деякий проміжок часу t не залежить від кількості вимог, які надійшли до системи, тобто не залежить від передісторії. Те, що проміжки часу не перетинаються, свідчить про взаємну незалежність протікання процесів у часі.

Потік є *ординарним*, якщо в один і той самий момент часу неможливе надходження двох або більше вимог.

Пуассонівський потік вимог є окремим випадком більш загального потоку Ерланга. Потік Ерланга γ -го порядку можна отримати шляхом просіювання пуассонівського потоку.

2.1.2. Організація черги

Дисципліни постановки вимог у чергу та вибору вимог із неї для обслуговування визначають порядок, за яким вимоги стають у чергу, якщо пристрій для обслуговування зайнятий, та порядок їх виходу з черги для обслуговування - якщо пристрій для обслуговування вільний.

Найпростіша дисципліна обслуговування передбачає поставлення вимог у чергу за порядком їх надходження. Вона має назву *перший прийшов - першим*

обслужили (ПППО), в англomовній літературі – FIFO (First Input First Output). Прикладом черги з такою дисципліною може бути черга до телефону-автомата.

Існує також інший спосіб організації черги, коли для обслуговування вибираються останні в черзі вимоги (*останній прийшов - першим обслужили* (ОППО)), в англomовній літературі – LIFO (Last Input First Output)). Цей спосіб також називається стеком або «магазинном». Прикладом черги з такою дисципліною обслуговування може бути паром, на якому перевозять авто, – автомобіль, який заїхав на паром перший, виїжджає з нього останнім.

Що стосується правила вибору вимог із черги, то вибір може бути *випадковим* (в англomовній літературі - RANDOM), наприклад вибір куль із барабана для гри в лото. Під час вибору вимог із черги може враховуватись їх пріоритет.

Черга може мати обмеження за довжиною або за часом перебування вимог у ній. Наприклад, якщо в черзі знаходиться більше трьох вимог, то нова вимога, яка надійшла, залишає систему, або вимога залишає систему, якщо час перебування її в черзі становить понад 2 хв. Прикладом черги з обмеженою кількістю місць є бункер, в який надходять заготовки, перш ніж їх буде оброблено верстатом. Буфери даних широко використовуються в комп'ютерній техніці. Під час обміну інформацією між пристроями, які мають різну швидкість обробки даних, інформація накопичується в буфері, а потім використовується пристроєм, що має меншу швидкість. Такі буфери організовуються в системах введення-виведення даних і мультиплексорах. У комп'ютерних мережах буфери створюють для організації черг повідомлень або пакетів.

Правила обслуговування вимог

Правила обслуговування вимог характеризуються часом обслуговування (розподілом часу обслуговування), кількістю вимог, які обслуговуються одночасно, і дисципліною обслуговування.

Обслуговування може бути організоване за допомогою одного або кількох ідентичних пристроїв. У першому випадку система називається *одноканальною*, у другому - *багатоканальною*. Час обслуговування вимог може бути детермінованим або заданим за ймовірнісним законом розподілу.

Якщо пристрої для обслуговування об'єднані в ланцюжок, то система називається багатofазовою, тому що вимоги в ній послідовно проходять кілька фаз обслуговування (наприклад, складальний конвеєр автомобільного заводу).

Дисципліна обслуговування визначає, за яких умов припиняється обслуговування вимог, як обирається для обслуговування наступна вимога, а також що станеться із частково обслуженою вимогою.

Розрізняють *безпріоритетні пріоритетні* дисципліни обслуговування. У разі безпріоритетного обслуговування порядок обслуговування визначається за дисципліною вибору вимоги з черги, наприклад ПППО. До безпріоритетних належить циклічна дисципліна обслуговування, яка часто використовується в комп'ютерних системах. Вимога (програма) багаторазово використовує пристрій для обслуговування (процесор) перед тим, як його залишити. Після закінчення кожного циклу обслуговування вимога знову надходить до черги для додаткового обслуговування.

Під час пріоритетного обслуговування для кожної вимоги задається деякий числовий параметр, значення якого визначає її *пріоритет*. Значення пріоритету

може бути незмінним (*статичний пріоритет*) або являти собою функцію, яка залежить від часу перебування вимоги в системі (*динамічний пріоритет*).

Пріоритет може бути також *відносним* або *абсолютним*. Відносний пріоритет передбачає, що надходження вимоги з вищим пріоритетом не перериває обслуговування менш пріоритетної вимоги (обслуговування без переривання). Вимоги з однаковими пріоритетами можуть утворювати черги.

Якщо в системі задається абсолютний пріоритет, то поява вимоги з більш високим пріоритетом перериває обслуговування менш пріоритетної вимоги (обслуговування з перериванням). У таких системах можуть утворюватися вкладені переривання, коли обслуговування вимоги, яка витиснула менш пріоритетну, буде перервано більш пріоритетною вимогою і т. д. Іноді в таких системах обмежують глибину переривання. Перервані вимоги можуть або залишати систему, або знову ставати в чергу для додаткового обслуговування.

Зрозуміло, що дисципліни обслуговування з абсолютними пріоритетами можуть використовуватись тільки для систем з одним пристроєм для обслуговування.

Вихідний потік вимог

Вихідний потік – це потік вимог, які залишають систему, до того ж вони можуть бути як обслуженими, так і необслуженими. Структура формування потоку вихідних вимог більш важлива для багатофазових систем, де вихідний потік одного пристрою для обслуговування (фази обслуговування) є вхідним для іншого. Ймовірнісні характеристики розподілу вимог вихідного потоку в часі залежать від щільності вхідного потоку та параметрів роботи пристроїв для обслуговування. З теорії масового обслуговування відомо, що вихідний потік вимог СМО з M пристроями з очікуванням для найпростішого вхідного потоку з параметром λ експоненціального розподілу часу обслуговування з параметром μ є найпростішим потоком. Це дає можливість аналізувати багатофазові системи і мережі СМО, в яких вихідний потік вимог одних систем обслуговування є вхідним для інших. У всіх інших випадках розподіл імовірності вихідних потоків вимог СМО має складнішу ймовірнісну природу і може вивчатись тільки шляхом спостереження за функціонуванням цих СМО під час моделювання.

2.2.Режими роботи системи масового обслуговування

На практиці часто доводиться вивчати режими роботи СМО, за допомогою яких описується деякий виробничий процес або система обробки інформації. Якщо в системі пристрої для обслуговування час від часу виходять з ладу, то вводиться поняття *режим відмови*. Під час дослідження деяких систем треба брати до уваги ще один режим – *блокування обслуговування*, пов'язаний з тимчасовим перериванням або сповільненням процесу обслуговування.

Зміна режиму роботи СМО може бути зумовлена зовнішнім впливом (наприклад, тимчасовою відсутністю деталей у технологічному процесі, ремонтом обладнання тощо) або виходом із ладу деякого пристрою системи (наприклад, блока живлення в комп'ютері).

2.3. Типи моделей систем масового обслуговування

У теорії систем масового обслуговування розглядаються тільки такі СМО, параметри ефективності яких можна отримати аналітично в замкненому або числовому вигляді. Для позначення таких моделей СМО часто використовують запис, запропонований Канделом - $X/Y/I$, де X - розподіл часу прибуття надходження вимог, Y - розподіл часу обслуговування, а I – кількість пристроїв для обслуговування.

Найпоширенішою моделлю, яка розглядається в теорії масового обслуговування, є модель типу $M/M/1$. Ця модель має тільки один пристрій для обслуговування (цифра 1), і в ній процеси розподілу часу надходження (перша буква M) та обслуговування (друга буква M) є марківськими. Для такої моделі час між двома надходженнями вимог до системи і час їх обслуговування мають експоненціальні розподіли. Модель типу $M/M/1$ може використовуватися, наприклад, для моделювання роботи однопроцесорної системи або стандартного пристрою для введення-виведення інформації (магнітного диска, принтера тощо).

У теорії масового обслуговування аналітичні результати отримано тільки для моделей типів $D/D/1$, $M/M/1$ і $M/G/1$. Для визначення характеристик моделей з іншими значеннями параметрів СМО потрібно використовувати методи імітаційного моделювання.

У реальних системах не завжди можна описати закони розподілу вхідних потоків вимог і часу їх обслуговування. Для їх визначення потрібно вміти оцінювати характер робочого навантаження системи. Зокрема, у разі моделювання комп'ютерної системи треба знати, коли до системи надходять нові завдання, скільки часу потрібно для виконання процесором кожного з них, як часто програма звертається до пристроїв для введення-виведення інформації. Для цього треба розробити діаграму роботи системи, на якій можна зобразити потоки вхідних завдань у систему, ресурси, до яких вони направляються, а також час обслуговування завдань на цих ресурсах.

Якщо графік робочого навантаження має періодичний характер, то під час спостереження за роботою комп'ютера є можливість отримати представницьку вибірку, яку можна застосовувати для аналізу показників ефективності СМО. Проте моделювання з використанням такого опису робочого навантаження (сценарію) дасть змогу відтворити тільки результати роботи комп'ютерної системи в минулому. На підставі цього не можна визначити, як працюватиме система в майбутньому.

Для того щоб спрогнозувати поведінку системи в майбутньому, потрібно виявити закономірності та визначити один або кілька розподілів імовірностей, а не використовувати необроблені дані, отримані шляхом вимірювання. Якщо для визначення робочого навантаження на систему використовуються розподіли ймовірності, а для аналізу результатів моделювання – відповідні

статистичні методи, то отримані результати можна поширити на більший діапазон робочих навантажень, ніж під час використання визначеного сценарію.

2.4. Основи дискретно-подійного моделювання систем масового обслуговування

Вивчаючи роботу СМО та розглядаючи алгоритми їх моделювання, можна визначити, як побудовані базові конструкції дискретно-подійних мов моделювання, наприклад такі, як генератори вимог, пристрої для обслуговування, термінатори вимог і хронологічні списки подій. Усі програмні засоби імітаційного моделювання дискретних систем включають засоби моделювання СМО будь-якої складності. Одна з перших мов імітаційного моделювання SP35 спочатку також розроблялась як мова моделювання СМО.

Під час моделювання СМО необхідно відтворити її роботу в модельному часі та організувати збір статистичних даних, потрібних для обчислення показників ефективності системи. Алгоритми моделювання можна побудувати, використовуючи формальні моделі, описані в розділі 1.8.3, тобто шляхом імітації станів модельованої системи. Дискретно-подійне моделювання ґрунтується на принципі просування модельного часу від події до події, якщо ці події упорядковані у модельному часі. Для реалізації цього принципу використовуються списки подій, де кожній події відповідає підпрограма обробки події, яка викликається в разі її настання.

Визначимо основні поняття і терміни, які використовуються під час моделювання СМО.

Об'єкт – будь-який елемент або компонент СМО, який має бути заданим явно в моделі СМО (наприклад, пристрій для обслуговування, клієнт, машина).

Список – множина (постійна або тимчасова) пов'язаних між собою об'єктів, упорядкована згідно з певним логічним правилом (наприклад, усі вимоги, які знаходяться в певний час у черзі, упорядковані за принципом «перший прийшов – першим обслужили» або за пріоритетами).

Подія – миттєва зміна стану системи, наприклад прибуття нової вимоги, або закінчення обслуговування вимоги в системі.

Повідомлення про подію – інформація про подію, яка сталась або станеться, і дані, необхідні для обробки події (запис про подію має включати інформацію про тип і час події).

Список подій – перелік намічених майбутніх подій, упорядкованих за часом їх виникнення, відомий також як список майбутніх подій (СМП).

Дія – операція, яка виконується протягом зазначеного проміжку часу (наприклад, час обслуговування або час між надходженнями вимог), для якої відомі час початку і закінчення (хоча цей час може бути визначено в термінах статистичного розподілу).

Затримка - тривалість невизначеного проміжку часу, для якого невідомо заздалегідь, коли він закінчується (наприклад, затримка вимоги в черзі за правилом «останній прийшов - першим обслужили», для якого початок обслуговування залежить від майбутніх надходжень).

Модельний час – позитивна зростаюча величина, яка відображає: перебіг часу в імітаційній моделі.

Годинник - змінна, яка відображає зміну модельного часу, у прикладах – *годинник (CLOCK)*.

Дискретно-подіте моделювання – це моделювання роботи системи в дискретні моменти часу, коли настають певні події, які відображають послідовність змін станів системи в часі. Розглянуті системи є динамічними, тобто змінюються в часі. Тому стан системи, властивості об'єкта і число активних об'єктів, параметрів, дій і затримок – функції часу, які постійно змінюються в процесі моделювання.

Для СМО з одним пристроєм для обслуговування події відбуваються в момент надходження вимоги до системи і в кінці її обслуговування пристроєм. Початок обслуговування – це умовна подія, яка залежить від стану пристрою (зайнятий або вільний) і числа вимог, що знаходяться в черзі. Затримку іноді називають умовним очікуванням, а дію - безумовним. Дії в такій системі характеризуються часом між надходженнями вимог і часом їх обслуговування пристроєм. Завершення дії – *первинна* подія, для керування якою в СМП уміщується повідомлення. Керування затримкою пов'язане з уміщенням об'єкта в інший список, який, можливо, відтворює чергу, де має місце затримка до того часу, коли умови, що склались у системі, дають змогу обробити вимоги. Закінчення затримки іноді називають *умовною* або *вторинною* подією, але такі події не зазначаються у відповідних повідомленнях про події та не з'являються в СМП.

2.5. Загальні відомості про мережі СМО

У загальному випадку мережу СМО можна зобразити у вигляді графа, вершинами якого є одноканальні або багатоканальні СМО (дуги визначають потоки пересування вимог).

Найпростіша мережа утворюється шляхом послідовного з'єднання кількох СМО (рис. 2.3). Таку мережу ще називають багатозафазовою СМО. Розрізняють *замкнені* та *розімкнені* мережі. Для замкненої стохастичної мережі не існує зовнішніх джерел вимог, тобто в ній завжди знаходиться однакова кількість вимог. Замкнена мережа ізольована від зовнішнього середовища. У розімкненій мережі (рис. 2.4) існують джерела і стоки вимог.

Найпростіша замкнена мережа, яку зображено на рис. 2.5, має тільки два вузли. Перший вузол містить M пристроїв для обслуговування, а другий - N . Така

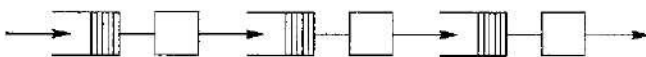
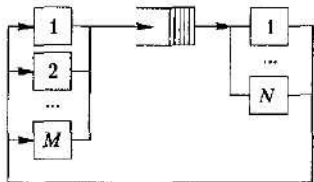


Рис. 2.14. Розімкнена мережа СМО

мережа є відомою моделлю СМО з відмовами та відновленням. Пристрої для обслуговування M можуть виходити з ладу та відновлюватись із заданими інтенсивностями у випадкові моменти часу. У цій мережі постійно знаходяться M вимог, які з'являються в разі відмови пристроїв обслуговування. Якщо пристрій виходить з ладу, до бригади з N ремонтниками надходить вимога на його ремонт, після завершення якої пристрій відновлює свою роботу. На рис.

це позначено зворотним зв'язком від N пристроїв. Дана мережа може використовуватись і для моделювання комп'ютерної системи, яка працює в режимі «запит-відповідь». У такій системі користувач не надсилає нового запиту до системи доти, доки не отримає відповіді на попередній запит. Запити обробляють будь-які з N комп'ютерів. Прикладами таких систем можуть бути автоматизовані системи продажу квитків, системи передавання транзакцій від касирів у банку та ін.

Схематичне зображення мережі СМО



Для розрахунків мереж СМО використовується теорія стохастичних мереж, яка ґрунтується на марківських або напівмарківських процесах, але більшість результатів отримано тільки для експоненціальних законів розподілу надходження та обслуговування вимог. Операційний аналіз, на відміну від теорії масового обслуговування базується на моделюванні логіки роботи системи. Це дає змогу встановити прості залежності між параметрами і показниками роботи системи, не абстрагуючись від процесів її функціонування.

Висновки

- ◆ Теорія масового обслуговування визначає залежність між параметрами потоків вимог, кількістю пристроїв обслуговування і їх продуктивністю, а також між режимами функціонування СМО та її ефективністю.
- ◆ Основні теоретичні результати теорії масового обслуговування отримано для систем, в яких процеси надходження та обслуговування вимог є марківськими та напівмарківськими.
- ◆ Замкнена мережа СМО – це ізольована від зовнішнього середовища мережа, для якої не існує зовнішніх джерел і стоків вимог.
- ◆ Розімкнена мережа СМО – це мережа, в якій вимоги, що обробляються пристроями обслуговування, можуть надходити із зовнішнього середовища або спрямовуватись до нього.
- ◆ Операційний аналіз є методом дослідження мереж СМО, в основі якого лежать операційні змінні. Основні результати операційного аналізу формуються у вигляді співвідношень між операційними змінними.
- ◆ Умови функціонування мережі СМО визначає гіпотеза про баланс потоків у мережі: кількість вимог, що надійшли до деякого вузла протягом тривалого періоду часу, дорівнює кількості вимог, які залишили цей вузол.

Контрольні запитання та завдання

1. Розгляньте потік Ерланга-го порядку. Як розподіляються проміжки часу між сусідніми вимогами? Поясніть, чому. До якого розподілу буде наближатись розподіл Ерланга у разі збільшення параметра γ ? Який розподіл матиме час надходження вимог, якщо γ прямує до нескінченності?

$$X_1 = X_3 = X_6 = \frac{U_3}{S_3} = 7,04 \text{ вимоги/с.}$$

2. Розгляньте об'єднаний потік вимог із попереднього завдання і припустіть, що тепер його потрібно розбити на кілька потоків. Нехай p – це ймовірність того, що деяка вимога з об'єданого потоку буде віднесена до підпотoku з номером k . Якщо інтенсивність сумарного потоку дорівнює L вимог за одиницю часу і ймовірності p вибираються довільно, то кожний з підпотоків створює пуассонівський процес з інтенсивністю Xp . Покажіть це.

3. Поясніть, чому система типу $B/B/1$ за умови $\rho = 1$ працює в сталому режимі й не має черги, а система типу $M/M/1$ за умови $\rho = 1$ переходить у нестационарний режим, і черга росте до нескінченності. Що буде за умови накладення обмежень на довжину черги для системи типу $M/M/1$? У цьому випадку при будь-якому співвідношенні значень для інтенсивності надходження вимог X і швидкості обслуговування μ ця система працюватиме в сталому режимі. Покажіть це.

4. З огляду на життєві ситуації наведіть кілька прикладів, які ілюструють роботу СМО. Розгляньте такі варіанти:

- а) вимоги надходять до системи обслуговування по одній;
- б) вимоги надходять до системи обслуговування пакетами;
- в) порядок надходження відомий заздалегідь;
- г) надходження повністю випадкове;
- д) джерело вимог невичерпане або обмежене; е) існує тільки одна черга або кілька черг;
- ж) час перебування в черзі обмежений;
- з) обслуговування виконується у дві фази.

Назвіть для наведених вами прикладів три-п'ять важливих характеристик, які потрібно було б розрахувати або виміряти для цих систем. За яких умов можна вважати вхідний потік вимог пуассонівським?

5. Опишіть, яким чином потрібно змінити роботу алгоритмів моделювання, щоб урахувати:

- а) обмеження, накладені на довжину черги;
- б) обмеження, накладені на час перебування в черзі;
- в) наявність m однакових пристроїв для обслуговування вимог.

Під час реалізації алгоритму для третього варіанта розгляньте визначення одного із вільних пристроїв у порядку наданих їм номерів $1, 2, \dots, m_i$ у випадковому порядку. Поясніть, чому для першого способу визначення вільних пристроїв їх завантаження буде різним.

6. Для розрахунків яких систем можна застосовувати операційний аналіз?

7. Сформулюйте гіпотезу про баланс потоків у мережі СМО. Поясніть, як її можна використовувати під час розрахунків різних СМО.

ТЕМА 3 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ

- 3.1. Доцільність використання імітаційного моделювання**
- 3.2. Поняття машинної імітації (імітаційного моделювання)**
- 3.3. Переваги та недоліки методу машинної імітації**
- 3.4. Загальна схема і цілі машинної імітації**
- 3.5. Методи проектування імітаційних моделей**
- 3.6. Програмна реалізація імітаційних моделей**
- Висновки**
- Контрольні запитання**

Завдяки зниженню вартості комп'ютерної техніки та доступності спеціальних програмних засобів методи імітаційного моделювання широко використовуються для дослідження складних об'єктів і систем у найрізноманітніших галузях людської діяльності.

3.1. Доцільність використання імітаційного моделювання

Переваги застосування імітаційного моделювання найбільш помітно виявляються у разі моделювання виробничих і технологічних процесів, процесів матеріально-технічного забезпечення виробництва, у логістиці, а також під час проведення бізнес-планування, екологічних і соціологічних досліджень. Важливо, що Імітаційне моделювання використовується, скоріше, як спосіб для осмислення проблеми і допомагає в цьому більше, ніж простий текстовий або математичний опис проблеми. Воно дає змогу глянути на складний процес ухвалення рішення більш масштабно, з погляду процесів, які відбуваються всередині системи, що моделюється. Часто моделювання припиняють ще до того, як будуть отримані конкретні результати. Визначення моменту, в який зацікавлені сторони зрозуміють, що ж насправді відбувається в системі, уже може бути рішенням проблеми. Тому навіть не завжди потрібно провадити статистичну обробку результатів експерименту. Звичайно, це не є правилом, адже імітаційні моделі взагалі використовуються саме для експериментальних цілей, але безсумнівно те, що Імітаційне моделювання – це технологічний процес, який проходить безліч стадій, вимагаючи від фахівців великих розумових і часових витрат.

1. Імітаційне моделювання дає змогу досліджувати внутрішні взаємодії у складних системах або підсистемах у межах складної системи, а також експериментувати з ними.

2. Моделюючи інформаційні, організаційні впливи і впливи зовнішнього середовища, можна оцінити ефекти цих впливів на поведінку (функціонування) системи.

3. На основі знань, отриманих під час проектування імітаційної моделі, можна визначити способи вдосконалення системи, яка моделюється.

4. Змінюючи вхідні дані під час моделювання і спостерігаючи за вихідними даними, можна виявити, які змінні найбільш важливі та як вони взаємодіють.

5. Імітаційне моделювання можна використовувати як метод для поліпшення рішень, отриманих під час аналітичного аналізу, а також для перевірки аналітичних рішень.

6. Імітаційне моделювання можна використовувати для проведення експериментів з новими проектами або стратегіями їх упровадження, щоб заздалегідь спрогнозувати результати.

7. Імітаційне моделювання можна застосовувати для визначення вимог, яким має відповідати пристрій або система.

8. Імітаційні моделі можна використовувати для навчання операторів складних технологічних процесів без зайвих затрат на придбання обладнання, яке може пошкоджуватись, і запобігаючи нещасним випадкам.

9. Для імітаційного моделювання можна використовувати засоби анімації, які дають змогу спостерігати за операціями, що моделюються.

10. Сучасне виробництво настільки складне, що взаємозв'язки в ньому можна інтерпретувати тільки шляхом проведення імітаційного моделювання.

Аналізуючи праці Р. Шенона і Дж. Банкса, можна визначити ситуації, коли провадити імітаційне моделювання не варто, а саме:

- ◆ проблему можна вирішити шляхом логічного аналізу ситуації;
- ◆ проблему можна розв'язати аналітичними методами, наприклад за допомогою теорії СМО;
- ◆ результати можна отримати шляхом проведення прямих експериментів з об'єктом без втручання в технологічний процес, наприклад за допомогою хронометражу на робочих місцях;
- ◆ для розроблення імітаційного проекту за визначений строк немає достатньої кількості ресурсів;
- ◆ не можна отримати необхідні вхідні дані (імітаційне моделювання потребує великої кількості різноманітних даних, які досить важко збирати, більш того, вони можуть бути просто недоступними);
- ◆ менеджери організації, яка замовляє проект, бажають отримати забагато від імітаційного моделювання і дуже швидко;

4- поведінка (режими функціонування) модельованої системи дуже складна або невизначена.

3.2.Поняття машинної імітації (імітаційного моделювання)

Динамічні процеси, які відбуваються у виробничих та економічних системах, настільки складні й різноманітні, що аналітичні методи дослідження операцій (математичне програмування, теорія масового обслуговування, теорія ігор і ін.), котрі на перший погляд здаються універсальними, насправді вельми нечасто придатні для прогнозування та аналізу фактичних ситуацій. Пояснюється це ось чим. Математичні моделі, які можуть бути реалізовані з допомогою ефективних обчислювальних методів, є надто спрощеними, а отже, не адекватні реальним процесам. Що ж до адекватних математичних моделей, то

їх здебільшого не можна реалізувати через труднощі обчислювального характеру. У такому разі часто є сенс застосувати машинну імітацію, що полягає в моделюванні на ПК реальної виробничої чи економічної системи.

Машинна імітація дає користувачеві змогу експериментувати з існуючими і створюваними системами тоді, коли на реальному об'єкті робити цього не можна або немає жодної рації. Такий метод набуває сьогодні особливої ваги, насамперед як інструмент удосконалення управління економікою та її галузями в межах усієї України. Є всі підстави стверджувати, що машинна імітація – один з найзагальніших і найпотужніших методів прикладного системного аналізу. Адже з допомогою імітаційних моделей системи розкривається сутність відповідних явищ і процесів за умови, що натурні досліди в реальному середовищі на реальних об'єктах виключаються.

З огляду на сказане можна дати таке означення імітаційного моделювання.

Імітаційне моделювання в широкому розумінні є процес конструювання моделі реальної системи та експериментування на цій моделі з метою визначити поведінку системи або оцінити (в рамках обмежень, зумовлених деяким критерієм чи сукупністю критеріїв) різні стратегії, що забезпечують функціонування даної системи.

У вужчому розумінні імітаційне моделювання – це відтворення на ЕОМ реальної виробничої чи організаційної системи. За такого тлумачення термін «імітаційне моделювання» має той самий сенс, що й «машинна імітація» або «машинне моделювання» (останні терміни відповідають експериментальному методу вивчення економіки з допомогою ЕОМ). Щоб застосувати такий метод для досліджень, створюють імітаційну систему, яка містить у собі *імітаційну модель*, а також внутрішнє і зовнішнє математичне забезпечення. До ЕОМ вводяться потрібні вхідні дані і спостерігаються зміни показників, котрі у процесі моделювання можуть аналізуватися й піддаватися статистичній обробці.

Імітаційна модель – це комплексна математична й алгоритмічна модель досліджуваної системи. А метод, що базується на розробці та дослідженні імітаційних моделей, називається машинною імітацією, або імітаційним моделюванням.

Термін «машинна імітація», який точніше відбиває сутність проблеми, у вітчизняній літературі використовується рідше, ніж термін «імітаційне моделювання»; у зарубіжній літературі першому терміну відповідають такі: **computer simulation** – комп'ютерне моделювання, **digital simulation** – системне моделювання, **digital simulation** – цифрове моделювання.

Слово «імітація» походить від англійського **imitate** – наслідувати, копіювати. Отже, термін «машинна імітація» означає копіювання на ЕОМ реальних чи гіпотетичних процесів.

Слово «моделювання» походить від французького **modeler** – ліпити (копію), тобто фактично воно також означає копіювати. Тому словосполучення «імітаційне моделювання» – тавтологія (копіювальне копіювання). Те саме стосується терміна «імітаційна модель». Зауважимо, проте, що в науковій літературі з інформатики й обчислювальної техніки дедалі частіше як синонім терміна «моделювання» вживається термін «симуляція».

Машинна імітація – це чисельний метод виконання на ЕОМ експериментів з математичними моделями, що описують поведінку складних систем протягом тривалих відтинків часу. Імітаційний експеримент принципово відрізняється від натурного, що провадиться не із самою реальною системою, а з її моделлю. Саме ця властивість імітаційного експерименту має вирішальне значення для дослідження складних економічних і виробничих систем, де на відміну від систем технічних і фізичних об'єктів реальні експерименти здебільшого нездійсненні.

3.3. Переваги та недоліки методу машинної імітації

Машинна імітація в усьому світі набула значного поширення при дослідженні складних систем завдяки важливим перевагам, що їх дістають користувачі цього методу. Розглянемо деякі з цих переваг.

Вдається відповісти на багато запитань, що постають на ранніх стадіях задуму і попереднього проектування систем, уникнувши застосування методу спроб і помилок, котрий пов'язаний зі значними витратами. Зокрема, на підставі аналізу імітаційної моделі можна заздалегідь визначити ефективність функціонування будь-якої спроектованої системи й попередити необгрунтовані витрати людських і матеріальних ресурсів на побудову нераціональних систем.

Метод дає змогу досліджувати особливості функціонування системи за будь-яких умов, зокрема й тих, котрі не реалізовані в натурних експериментах. При цьому параметри системи і навколишнього середовища можна варіювати в як завгодно широких межах, відтворюючи довільну обстановку. Завдяки такому підходу різко зменшується потреба в складному лабораторному обладнанні та експлуатаційних випробуваннях системи. Наприклад, під час польоту американського космічного корабля «Аполлон-13» машинне моделювання застосували для аналізу надзвичайних заходів, перш ніж було подано команди щодо їх здійснення. Це дозволило космонавтам благополучно повернутися на Землю після вибуху балона з киснем на борту корабля.

Стає можливим прогнозувати поведінку системи в близькому та віддаленому майбутньому, екстраполюючи на моделі результати промислових випробувань. У такому разі дані, здобуті раніше, поповнюються завдяки застосуванню статистичного підходу.

Імітаційні моделі технічних і технологічних систем та пристроїв дають змогу в багато разів скоротити час їх випробування: від днів і місяців у реальних умовах до секунд і хвилин на ПК. Єдиним обмеженням тут можуть бути наявні ресурси машинного часу і пам'яті.

За допомогою методу машинної імітації можна штучним шляхом швидко й у великому обсязі дістати потрібну інформацію, що відбиває хід реальних процесів, уникнувши дорогих, а часто й неможливих натурних випробувань цих процесів – відповідні результати через обмежений обсяг вибірки здебільшого не зовсім достовірні (вірогідні).

Імітаційна модель є надзвичайно гнучким пізнавальним інструментом, здатним відтворювати довільні як реальні, так і гіпотетичні ситуації, оскільки на неї не поширюються жодні фактичні обмеження.

З огляду на те, що дослідження і оптимізацію деяких складних економічних систем не можна виконати ні з допомогою лабораторних чи натурних експериментів, ні аналітичними методами, імітаційне моделювання на ЕОМ часто буває єдиним реалізовуваним способом розв'язування таких задач.

Зауважимо, проте, що метод машинної імітації, попри всі його переваги та універсальність, прийнятний аж ніяк не завжди, оскільки виконання розрахунків на імітаційних моделях потребує значних витрат часу дослідників, програмістів та ПК.

Машинну імітацію як чисельний машинний метод розв'язування складних задач доцільно застосовувати за таких умов:

- непридатність або відсутність аналітичних методів розв'язування задач (правило – "коли інші методи безсилі");
- цілковита упевненість в успішному створенні імітаційної моделі, яка адекватно описує досліджувану систему (процес), зокрема в тому, що вдасться зібрати всю необхідну інформацію про модельовану систему (процес), забезпечивши вірогідну імітацію на ПК реальних ситуацій (будувати імітаційну модель стохастичних процесів, коли не можна дістати опис потрібних характеристик випадкових величин і подій, – марний замір);
- можливість використати сам процес побудови імітаційної моделі для попереднього дослідження модельованої системи з метою напрацювання рекомендацій щодо поліпшення умов її функціонування. З такою ситуацією стикаємося і при створенні проєктів КСУ: витрати на передпроектне обстеження об'єкта комп'ютеризації дуже часто компенсуються за рахунок економічного ефекту від реалізації висновків, зроблених на підставі матеріалів обстеження, у плані вдосконалення діючої системи без застосування засобів комп'ютеризації.

Коли приймають рішення щодо фінансування того чи іншого впровадження машинної імітації, обов'язково зіставляють сподіваний економічний ефект з відповідними розрахунковими витратами. І тут потрібна певна обережність з висновками. Напевне, є рація додержувати зарубіжних рекомендацій: створювати імітаційну модель доцільно лише в тому разі, коли сподіваний вигаш у грошовому еквіваленті перебиває обсяг витрат більш ніж у 10 раз. Адаже вартість робіт, як показує практика, занижується втричі, тоді як сподіваний ефект у стільки ж разів завищується.

3.4. Загальна схема і цілі машинної імітації

У процесі машинного моделювання на ПК із певною достовірністю відтворюються реальні ситуації. Зауважимо, що іноді ситуації «програються», тобто реально імітуються досліджувані дії. Наприклад, при військових навчаннях відтворюються бойові дії з максимальним наближенням до ймовірних ситуацій воєнного часу. Значного поширення в навчальному процесі підготовки та перепідготовки кадрів набули ділові ігри, учасники яких за наперед розробленим сценарієм імітують у лабораторних умовах ту чи іншу виробничу ситуацію. Ці методи, безперечно, корисні в навчальному плані. Проте для

розв'язування складних економічних задач і задач організаційного управління імітація реальних дій в натурних умовах неможлива або пов'язана зі значними матеріальними витратами.

Проблеми такого плану доцільніше розв'язувати, подаючи складну функціональну систему з допомогою логіко-математичної моделі, занесеної в ПК. При цьому фактори невизначеності, динамічні характеристики та весь комплекс взаємозв'язків між елементами досліджуваної системи набирають вигляду формул, котрі зберігаються в пам'яті машини.

Імітацію системи починають з деякого цілком конкретного – початкового стану. У результаті прийнятих рішень, а також унаслідок настання ряду контрольованих подій (серед них можуть бути й випадкові) система в наступні моменти часу переходить до інших станів. Еволюційний процес триває так доти, доки не настане кінцевий момент планового періоду.

Відтинки (відрізки) часу внутрішньопланового періоду нерідко бувають чітко визначеними й утворюють упорядковану послідовність на досить великому проміжку імітування. Тому імітаційний експеримент пов'язаний з величезною кількістю обчислень на потужних ПК. Відображення реального тривалого процесу на ПК за кілька хвилин чи секунд називається *стискуванням часу*.

Розглянемо можливі цілі створення імітаційної моделі, призначеної для вивчення проблем організаційного управління.

Вивчення діючої функціональної системи.

Аналіз гіпотетичної функціональної системи.

Проектування досконалішої системи. При дослідженні складних економічних систем на імітаційних моделях насамперед слід установити *адекватність* моделі реальним об'єктам. У разі неадекватності моделі дослідник ризикує дістати недостовірні результати, а на їх підставі прийти до помилкових висновків. Тому оцінювання адекватності моделі – обов'язковий етап моделювання, котрий сам по собі може бути великою і складною задачею.

Перевірку достовірності моделі називають її *верифікацією* (від лат. *verus*– істинний і *ficatio (facio)* – роблю).

Адекватна (від лат. *adaquatus*– прирівнюваний) імітаційна модель математично і логічно з певною мірою наближення відображає досліджувану систему. Логічні елементи моделі відповідають операціям, виконуваним у реальній дійсності, а математичний опис визначає функції, що реалізуються в реальній системі. Імовірнісні оператори адекватної імітаційної моделі відображають випадковий характер подій реальної системи. Ендогенні параметри моделі при відповідних вхідних чинниках мають бути інформативними, тобто давати вірогідні повідомлення про систему.

Оцінювання адекватності моделі передбачає *оцінювання адекватності принципової структури моделі та оцінювання достовірності її реалізації*.

Верифікувати імітаційну модель реальної системи вельми складно. Зробити це можна з допомогою або спеціально дібраних конкретних прикладів, котрі не обов'язково повинні містити реальну інформацію, або реальних задач, для яких відомі розв'язки, здобуті іншими способами.

3.5. Методи проектування імітаційних моделей

Перш ніж розпочати побудову моделі, потрібно мати певну схему її проектування, за якою визначають основні принципи і методи розроблення імітаційної моделі. Сукупність правил виявлення та застосування системних принципів і методів визначає методологію проектування. Її можна розглядати на різних рівнях деталізації залежно від вибраних засобів розроблення програмних реалізацій імітаційної моделі. За допомогою вибраних програмних засобів визначають і можливі методи їх застосування. Так, наприклад, вибір за основу імітаційної моделі мереж СМО або Петрі заздалегідь визначає метод побудови її у вигляді формальних схем цих мереж.

Для створення імітаційних програм на рівні мовних засобів побудови моделей потрібно розробити алгоритми імітації, які можна подати у вигляді наборів типових обчислювальних схем. Під обчислювальною схемою імітаційного алгоритму розуміють спосіб його організації, який дає змогу відтворити в модельному часі динаміку функціонування системи.

Отже, перш ніж розпочати проектування імітаційної моделі, необхідно вибрати засоби програмування. Однак існують загальні методи побудови програмних реалізацій та проектування імітаційних моделей, які не залежать від вибраних програмних засобів.

Під час проектування імітаційної моделі варіантний метод є найпростішим та широко застосовуваним. Проектувальник послідовно крок за кроком створює імітаційну модель, опираючись тільки на свій досвід та інтуїцію. У процесі проектування розглядаються кілька варіантів кожної частини модельованої системи для її відображення в імітаційній підмоделі. Найдоцільніший варіант вибирається з урахуванням рішень, прийнятих відносно інших частин модельованої системи. Це так звана послідовна схема проектування, згідно з якою вибір варіанта імітаційної моделі є суб'єктивним і залежить від рівня знань проектувальника про систему.

Застосування варіантного методу рідко приводить до допустимих проектних рішень і не відповідає загальній схемі системного аналізу імітаційного моделювання. Така схема передбачає виконання ітераційної процедури, під час якої проектувальник не один раз повертається до вже розроблених частин імітаційної моделі і коригує їх, доки не буде впевненим, що модель відповідає цілям моделювання, або не відмовиться від неї.

Ітераційний метод

Суть цього методу полягає в тому, що шляхом багатьох ітерацій спроектована спочатку імітаційна модель перетворюється в таку, яка відповідає цілям моделювання. Цей метод є методом «проб і помилок», що передбачає послідовні циклічні зміни, у результаті чого отримують модель, яка задовольняє вимогам точності та адекватності. Циклічний ітераційний метод проектування потребує розгляду послідовності процедур прийняття рішень у процесі проектування. Крім того, весь хід проектування та остаточний результат значною мірою залежать від вибору початкової імітаційної моделі. Основна проблема в разі застосування як ітераційного, так і варіантного методу

проектування полягає у виборі початкового варіанта моделі. Через те, що вже під час формулювання проблеми та в процесі змістовної постановки задачі висуваються вимоги до моделі, визначаються вхідні та вихідні дані, проєктувальник повинен вибрати початкову модель, використовуючи метод аналогії, який базується на знанні характеристик компонентів системи, технологічних засобів і прийнятих рішень у подібних умовах. Вибір вихідної імітаційної моделі дуже впливає на результати проєктування та може зробити його неможливим або занадто дорогим. Визначення рівнів точності, достовірності й правильності вибраної імітаційної моделі є самостійною проблемою моделювання, яку необхідно вирішувати під час розроблення моделі.

Методи внесення змін у модель базуються на принципі напрямленого дослідження. Для його застосування можна побудувати в просторі параметрів імітаційної моделі гіперповерхню її показників точності та оптимізувати або хоча б поліпшити ці показники. Сама ж процедура внесення змін у варіант моделі звичайно потребує перевірки гіпотез, які формулюють з огляду на результати проєктування попередніх моделей.

Якщо результати порівняння моделі і реальної системи незадовільні, то перш ніж вносити зміни в модель, необхідно сформулювати ряд гіпотез, за допомогою яких можна визначити причину невідповідності.



Гіпотези доцільно формулювати для кількох рівнів представлення імітаційної моделі :

- ◆ опису структури;
- ◆ алгоритмів поведінки;
- ◆ параметрів і вхідних даних.

Вибір рівня, на якому коригуватиметься модель та локалізуватимуться причини невідповідності, є скоріше мистецтвом, ніж наукою, і успішний результат залежить від досвіду, знань та інтуїції проєктувальника. Пошук причин невідповідності потрібно починати на рівні вхідних даних, для чого оцінюють чутливість моделі до їхніх змін. Якщо виявилось, що незначна зміна вхідних даних спричиняє значну зміну вихідних, то необхідно уточнити вхідні дані для моделі і (або) локалізувати блоки моделі, на які найбільше впливають ці вхідні дані. Виявлення причин такої сильної залежності може потребувати зміни структури імітаційної моделі шляхом заміни окремих блоків моделі на більш деталізовані, що, у свою чергу, спричинить зміну внутрішніх параметрів

моделі та алгоритмів функціонування. Отже, у цьому разі рівні, на яких вносяться зміни в імітаційну модель, є взаємопов'язаними.

Параметричне налагодження імітаційної моделі вимагає пошуку найкращих (оптимальних) параметрів, при яких ступінь невідповідності між моделлю та системою буде мінімальним. Це типове завдання оптимізації параметрів моделі.

Алгоритми поведінки моделі можуть змінюватись локально, для окремих блоків моделі, або для моделі в цілому. Такі зміни вимагають більш детального вивчення поведінки модельованої системи і можуть змінити рівень деталізації в моделі.

Змінити структуру моделі складніше, ніж налагодити параметри моделі, бо це може спричинити зміну алгоритмів поведінки, параметрів і вхідних даних моделі. Таку перебудову моделі можна починати тільки тоді, коли всі інші можливості вичерпано. Перебудова структури моделі може призвести до глобальних змін імітаційної моделі та її заміни новою. Тому перш ніж змінювати структуру моделі, необхідно перевірити всі гіпотези щодо витрат, які потрібні для зміни моделі. Починати перевірку слід з гіпотези, яка вимагає мінімальних витрат, а отже, і мінімальних змін імітаційної моделі.

3.6. Програмна реалізація імітаційних моделей

Програму для ПК можна розробити двома способами:

- 1) звичайними засобами програмування із застосуванням проблемно-орієнтованих або машинно-орієнтованих мов (C);
- 2) з допомогою спеціалізованих мов моделювання.

Перший спосіб використовується, коли імітаційна модель не дуже складна, застосовується нечасто і програмується спеціалістами, котрі не мають досвіду роботи з імітаційними моделями. Проте при цьому програмістові доводиться заново складати підпрограми стандартних процедур, використовуваних у всіх імітаційних моделях (генерування випадкових змінних, статистична обробка даних, розміщення інформації всередині машинної пам'яті, складання основної програми, яка забезпечує правильну черговість подій та просування імітаційного процесу по осі часу).

Отже, застосування універсальних мов програмування має і переваги (мінімум обмежень на вхідний формат, значна поширеність), і недоліки (чималі витрати часу на програмування та налагодження програм). Створювати програмне забезпечення імітаційного моделювання допомагають спеціалізовані машинні мови. При їх використанні достатньо лише задати функцію розподілу ймовірностей. Тоді автоматично генеруються випадкові події за даним законом розподілу. Деякі зі спеціалізованих програм забезпечують збір статистичних даних за тими чи іншими досліджуваними характеристиками імітаційної системи і видачу результатів машинного моделювання в наперед заданій формі. За допомогою таких програм упорядковують події та реєструють у часі кожний перехід системи з одного стану до іншого.

Зауважимо, що при машинному моделюванні спеціалізовані мови порівняно з універсальними мають такі переваги:

менші витрати на програмування;

ефективніші методи виявлення помилок;
стислість і точність вираження понять, котрі характеризують імітаційні процеси;
можливість заздалегідь створювати для користувачів стандартні підпрограми, що можуть використовуватися в будь-якій імітаційній моделі;
автоматичне формування певних типів даних;
зручність нагромадження та подання зображення даних;
забезпечення керування та контролю за розподілом машинної пам'яті в ході імітаційного моделювання.

Мови моделювання реалізуються, як правило, з допомогою універсальної *провідної програми*. Під її керівництвом виконується кожна імітаційна модель. Провідна програма автоматично реалізує типи для імітації службові операції:

- переміщує в потрібні моменти системний час;
- реєструє плановані події;
- організує реалізацію подій, що настають;
- динамічно розподіляє пам'ять обчислювальної машини і т.ін.

Спираючись на перелічені можливості провідної програми, користувач мови моделювання може розробляти модель, знаючи таке: системний час рухається автоматично; будь-яка запланована подія відбудеться за відповідних умов; довільна кількість одночасних подій може бути реалізована тощо. Отже, можна сконцентрувати всю увагу на опису об'єктів системи, не завантажуючи модель допоміжними елементами.

Зауважимо, що попри істотні переваги спеціалізованих мов моделювання обмежитися лише їх використанням (орієнтуватися тільки на спеціалізовані програми) не можна, і насамперед з таких причин.

1. Щоб використовувати деякі мови імітаційного моделювання, потрібно досконало знати пов'язані з ними проблемно-орієнтовані мови.

2. Не завжди обчислювальні машини, які встановлено в місцях їх використання, забезпечені трансляторами потрібних мов моделювання, а придбати такі транслятори через їх високу ціну мають змогу не всі.

3. У мовах імітаційного моделювання встановлені жорсткі обмеження на вихідні формати, що нерідко створює незручності для користувачів.

4. Написані спеціалізованими мовами імітаційні моделі не достатньо гнучкі, і моделювання через неоптимальність моделі потребує значних витрат машинного часу та великого обсягу машинної пам'яті. Отже, постановка серії експериментів може бути економічно не вигідною. Не виключається навіть така ситуація, коли модель не «вміщується» в пам'яті ПК.

Нині використовуються численні (понад 500) мови моделювання. Їх поділяють на такі класи.

Мови моделювання неперервних процесів (динамо, 360/SYSTEM, CSMPit.ін.).

Мови моделювання неперервно-дискретних процесів (недіс, DISLINit.ін.).

Мови моделювання дискретних процесів (GPSS – General Program System Simulation).

Розрізняють мови імітаційного моделювання за такими ознаками:

- 1) способом обліку часу подій, що відбуваються, та виконуваних дій;
- 2) правилами найменування структурних елементів;
- 3) способом перевірки умов, при яких реалізуються дії;
- 4) видами статистичних досліджень, які можливі за наявності даних;
- 5) труднощами, що виникають при зміні структури моделі.

Для моделювання дискретних процесів, характерних систем організаційного управління, створено мови, що зорієнтовані на дії (**CSL, ESP**), *події* (симскрипт, симком, симпак, **GASP** – базовою мовою програмування є фортран), *процеси* (**SIMULA, SOL, ASPOL, симула-67** – базовою мовою програмування є алгол), *потоки повідомлень – транзакти* (**GPSSV, GPSS/PC, CSS, BOSS** – мають власні мовні засоби), *потоки вимог – активатори* (симдіс, **SMPL** – базова мова програмування **PL/I**), *агрегати* (**AIC**, сапас – мають свої мовні засоби). Найбільшого поширення в нашій країні набули мови типу **GPSS**. Сучасний варіант цієї мови **GPSS/PC**, зорієнтований на інтерактивне використання, описаний в цьому посібнику.

Висновки

◆ Імітаційне моделювання можна застосовувати як універсальний підхід для прийняття рішень в умовах невизначеності та для врахування в моделях факторів, які важко формалізуються.

◆ Імітаційне моделювання включає такі взаємопов'язані етапи:

формулювання проблеми і змістовна постановка задачі; розроблення концептуальної моделі; розроблення і програмна реалізація імітаційної моделі; перевірка правильності та достовірності моделі; організація та планування проведення експериментів; прийняття рішень за результатами моделювання.

◆ У змістовній постановці задачі формулюються цілі моделювання, описуються основні характеристики системи, вхідні та вихідні змінні, їх взаємозв'язок, зовнішні впливи на систему, а також визначаються основні критерії функціонування системи та обмеження.

◆ Концептуальною називається абстрактна модель, яка виявляє причинно-наслідкові зв'язки, властиві досліджуваному об'єкту в межах, визначених цілями дослідження. Це формальний опис об'єкта моделювання, який відображає концепцію (погляд дослідника на проблему).

◆ Під формалізацією розуміють такий опис моделі, який припускає використання математичних методів дослідження, тобто на завершення цього етапу розробляється логіко-математичний опис модельованої системи з урахуванням динаміки її функціонування.

◆ Мови моделювання залежно від способу визначення внутрішньосистемного часу традиційно поділяють на три групи: неперервні, дискретні та неперервно-дискретні (комбіновані).

◆ Терміном *модельний час* називають арифметичну величину, яка має додатні зростаючі значення та під час моделювання відображає плин часу в моделі.

◆ Побудова діаграми подій – ефективний проміжний етап перетворення концептуальної моделі в програму імітаційної моделі; він дає змогу

здебільшого пропускати етап алгоритмізації моделі та переходити безпосередньо до розроблення програми.

◆ Обґрунтування адекватності моделі доводить, що модель, у межах сфери застосування працює із задовільною точністю, сумісною з метою моделювання.

Контрольні запитання та завдання

1. Які засоби формалізованого зображення можуть використовуватись для концептуальних та імітаційних моделей? Наведіть приклади.

2. Якими параметрами імітаційної моделі можна характеризувати вибраний рівень деталізації? Чому детальніша модель дорожча? Чому вона більш стійка до змін вхідних даних?

3. Перерахуйте основні затрати на розроблення та експлуатацію імітаційної моделі.

4. Як можна відобразити структуру імітаційної моделі? Чи залежить структура моделі від вибраних засобів моделювання?

5. У чому полягають істотні відміни моделювання, орієнтованого на події та процеси? Наведіть приклади подій та процесів для СМО.

6. Які проблеми виникають під час розробки засобів автоматизації побудови моделей? Наведіть види цих засобів та назвіть їх переваги.

7. Від чого залежить точність результатів моделювання? Проаналізуйте цю проблему, починаючи від етапу збору вхідних даних для імітаційної моделі. Як пов'язані між собою точність і адекватність моделі?

ТЕМА 4 МЕТОД МОНТЕ-КАРЛО

4.1. Означення та теоретичні основи методу Монте-Карло.

4.2. Суть методу.

4.3. Основні етапи методу.

4.1. Означення та теоретичні основи методу Монте-Карло.

Метод Монте-Карло – загальна назва групи числових методів, що ґрунтуються на одержанні великої кількості реалізацій стохастичного (випадкового) процесу, який формується у той спосіб, щоб його ймовірнісні характеристики збігалися з аналогічними величинами задачі, яку потрібно розв'язати.

Метод Монте-Карло - чисельний метод вирішення математичних задач за допомогою моделювання випадкових чисел.

Метод Монте-Карло – це метод імітації для приблизного відтворення реальних явищ. Він об'єднує аналіз чутливості (сприйнятливості) і аналіз розподілу ймовірностей вхідних змінних. Цей метод дає змогу побудувати модель, мінімізуючи дані, а також максимізувати значення даних, які використовуються в моделі. Побудова моделі починається з визначення функціональних залежностей у реальній системі. Після чого можна одержати кількісний розв'язок, використовуючи теорію ймовірності й таблиці випадкових чисел.

Метод Монте-Карло широко використовується у всіх випадках симуляції на ПК.

Теоретичною основою методу статистичного моделювання є закон великих чисел.

У теорії ймовірностей **закон великих чисел** ґрунтується на доведенні низки теорем для різних умов збіжності за ймовірністю середніх значень результатів (на підставі великої кількості спостережень) до деяких величин.

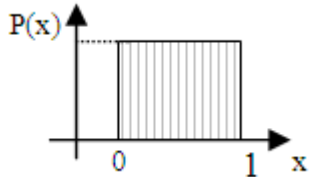
Розв'язування задач методом статистичного моделювання **полягає в такому:**

- ❖ опрацювання й побудова структурної схеми процесу, виявлення основних взаємозв'язків;
- ❖ формалізований опис процесу;
- ❖ моделювання випадкових явищ (випадкових подій, випадкових величин, випадкових функцій), що притаманні досліджуваній системі;
- ❖ моделювання процесу функціонування системи (на підставі використання даних, що отримані на попередньому етапі) – відтворення процесу відповідно до розробленої структурної схеми і формалізованого опису (імітаційні прогони);

4.2. Суть методу.

Суть методу полягає в тому, що посередництвом спеціальної програми на ПК виробляється послідовність псевдовипадкових чисел з рівномірним законом розподілу від 0 до 1. Потім дані числа за допомогою спеціальних програм перетворюються в числа, розподілені згідно із законом Ерланга, Пуассона, Релея і т.д.

Отримані таким чином випадкові числа використовуються як вхідні параметри економічних систем :



При багатократному моделюванні випадкових чисел, які ми використовуємо як вхідних параметрів системи (моделі), визначаємо математичне очікування функції $M(Q)$ і, при досягненні середнім значенням функції Q рівняння не нижче заданого, припиняємо моделювання.

Статистичні випробування (метод Монте-Карло) характеризуються основними параметрами:

D - задана точність моделювання;

P - вірогідність досягнення заданої точності;

N - кількість необхідних випробувань для отримання заданої точності із заданою вірогідністю.

Визначимо необхідне число реалізацій N , тоді $(1 - D)$ буде вірогідність того, що при одному випробуванні результат не досягає заданої точності D ;

4.3. Основні етапи.

Метод Монте-Карло складається з 4 етапів:

1) Побудова математичної моделі системи, що описує залежність характеристики системи, яка моделюється, від значень випадкових факторів.

2) Визначення закону розподілу ймовірностей для випадкових факторів.

3) Визначення діапазону значень для кожного випадкового фактору, генерація значень випадкових факторів у визначеному діапазоні, імітація поведінки системи шляхом багаторазового проведення імітаційних експериментів і отримання оцінки характеристики системи, що моделюється (узагальнення результатів експериментів).

4) Оцінка точності отриманих результатів.

2. Приклад.

Необхідно виконати імітаційне моделювання ризиків інвестиційного проекту з виробництва деякого продукту.

а. Ключовими варійованими параметрами є: змінні витрати, обсяг випуску і ціна.

b. Вважаємо, що всі ключові параметри підкоряються одному закону розподілу ймовірностей.

c. Задані діапазони можливихзм інварійованих параметрів.

d. Інші параметри проекту вважаються постійними величинами протягом строку реалізації проекту.

$$NPV = \sum_{t=1}^n \frac{NCF_t}{(1+r)^t} - I_0$$

$$NCF = [Q \cdot (P - V) - F - A] \cdot (1 - T) + A$$

Ключові варійовані параметри проекту з виробництва продукту

Показники	Сценарій		
	Найгірший	Найкращий	Вірогідний
Обсяг випуску – Q	150	300	200
Ціна за штуку – P	40	55	50
Змінні витрати – V	35	25	30

Незмінні параметри проекту з виробництва продукту

Показники	Найбільш вірогідне значення
Постійні витрати – F	500
Амортизація – A	100
Податок на прибуток – T	60%
Норма дисконту – r	10%
Строк реалізації проекту – n	5
Початкові інвестиції – I_0	2000

Генерація значень варійованих параметрів проекту
в MS Excel

Хід роботи:

Перед початком роботи було налаштовано середовище для розрахунків завдяки додаванню пакета для аналізу (рис.1).

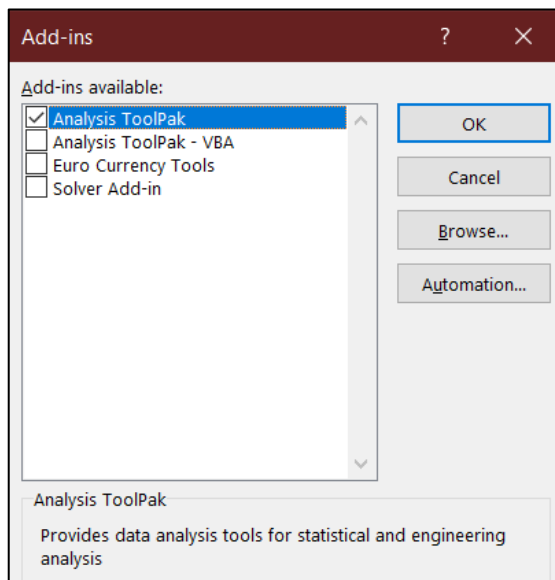


Рис.1. – Додавання необхідних пакетів для проведення аналізу

Генерація випадкових чисел виконувалася наступним чином як показано на рисунку 2.

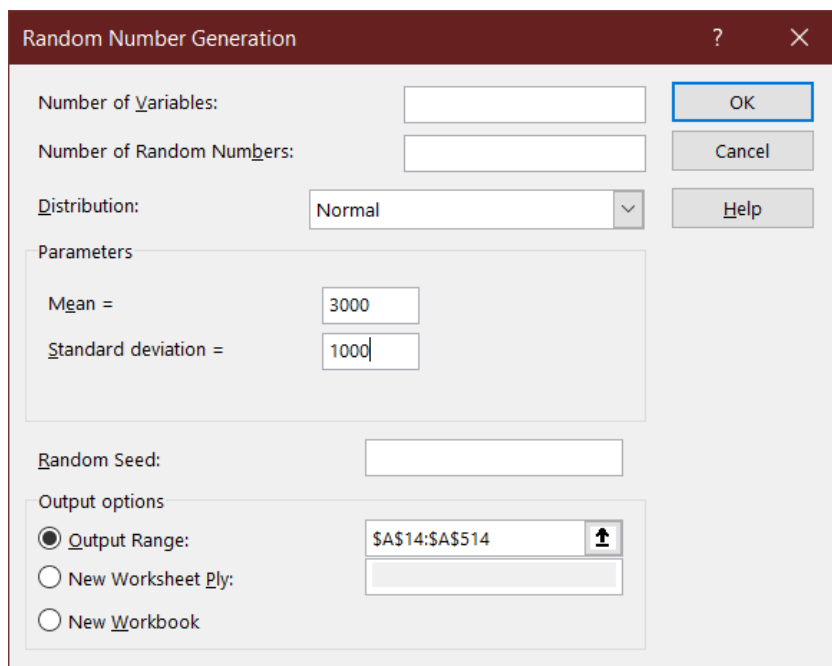


Рис.2. – Генерація випадкових змінних витрат проекту за допомогою інструментів графічного інтерфейсу Excel

Таблиця 1.

Ключові змінні проєкту з виробництва продукту

	Вірогідни й	Вірогідний	Найкращий
Обсяг реалізації	11716	11997	12289
Ціни за одиницю	2644	3000	3536
Змінні витрати на 1 продукції	1249	891	523

Оплата праці на одиницю продукції	1386	900	529
-----------------------------------	------	-----	-----

Аналіз результатів імітаційного експерименту

Імітаційний аналіз:

Нижче, на рисунках показано процес виконання імітаційного аналізу за методом Монте-Карло.

Імітаційний аналіз (Метод Монте-Карло) Розподілення з рівною імовірністю					
Початкові інвестиції (I)	1000000	Норма [®]		0.01	
Постійні витрати (F)	11661155	Налог (T)		0.05	
Амортизація (A)	2000	Срок (n)		3	
Показники	Змінні (V)	Кількість (Q)	Ціна (P)	Надходження (NCF)	NPV
Середнє значення	717.40	10997.66	4012.80	23351717.63	67677056.1
Стандарт. Відхилення	270.15	565.54	1187.98	-10584880.17	-32129976
Коеф. Варіації	0.38	0.05	0.30	-11077997.25	-33580226.05
Мінімум	250.00	10000	2006	5604002.75	15481289.19
Максимум	1193.00	11995	5982	43493855	126914784.2
Число випадків NPV<0					0
Сума збитків					-19444617.49
Сума прибутків					15860564.18
Імовірність p (NPV<=X)			Вел.(X)	Норм.(X)	p(NPV<=X)

Рис.3. – Імітаційний аналіз (Метод Монте-Карло) розподілення з рівною імовірністю

Ефективніше себе показав імітаційний аналіз із нормальним розподіленням (рис.4).

Імітаційний аналіз (Метод Монте-Карло) Нормальне розподілення					
Початкові інвестиції (I)	1000000	Норма (r)		0.01	
Постійні витрати (F)	11661155	Налог (T)		0.05	
Амортизація (A)	2000	Срок (n)		3	
Показники	Змінні (V)	Кількість (Q)	Ціна (P)	Надходження (NCF)	NPV
Середнє значення	890.90	11996.67	2999.73	12955903.84	37103121.53
Стандарт. Відхилення	134.34	99.04	102.12	-11081028.38	-33589140.6
Коеф. Варіації	0.15	0.01	0.03	-11077997.25	-33580226
Мінімум	522.91	11716	2644	12525516.47	35837358.64
Максимум	1249.35	12289	3536	15622852.22	44946577.29
Число випадків NPV<0					0
Сума збитків					-7309341.07
Сума прибутків					21480045.48
Імовірність p (NPV<=X)			Вел.(X)	Норм.(X)	p(NPV<=X)

Рис.4. – Імітаційний аналіз (Метод Монте-Карло) нормальне розподілення

ТЕМА 5. ОСНОВИ МОДЕЛЮВАННЯ ПОДІЙ

5.1. Моделювання подій сигналів.

5.2. Кінцеві автомати. Моделювання поведінки спільно об'єктів.

5.3. Процеси і потоки керування.

5.4. Моделювання систем реального часу

5.1. Моделювання подій сигналів.

Реальний світ насичений подіями, які дуже часто настають раптово та одночасно. В UML будь-яке явище, яке може мати місце в дійсності, моделюється як подія.

Усі системи, пов'язані з реальністю, тією чи іншою мірою є динамічними, цю динаміку зумовлюють саме події, що відбуваються всередині системи або за її межами. Як наприклад, роботу банкомата ініціює користувач, який натискає кнопку для здійснення транзакції та видачі готівки.

Подія (event) – це опис істотного факту, що відбувся в певному часі й просторі. Події можуть бути синхронними й асинхронними; їхнє моделювання – одна зі складових моделювання процесів і потоків.

Сигнал (signal) є різновидом події, при використанні якого повідомлення передається асинхронно від одного екземпляра до іншого. Події можуть бути внутрішніми або зовнішніми. Зовнішні події передаються між системою, а внутрішні – між об'єктами, що існують у самій системі.

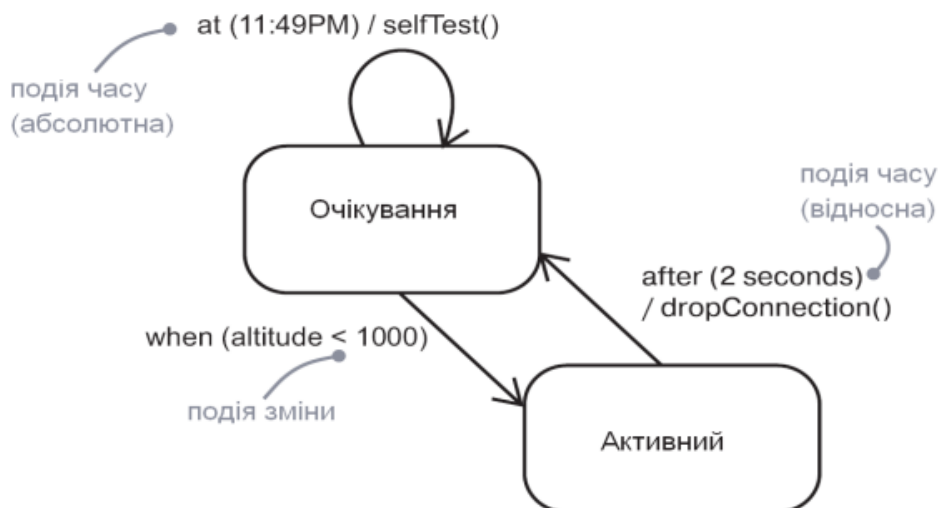
В UML можна моделювати чотири види подій

1. Сигнал. Подія сигналу є його екземпляр (як сам факт передавання сигналу).

2. Виклики. Подією виклику є отримання запиту об'єктом на виконання операції над ним

3. Закінчення проміжку часу. Подія часу є закінченням якогось проміжку часу. В UML така подія моделюється за допомогою ключового слова after

4. Зміна стану. Ця подія описує зміну стану або виконання деякої умови. В UML подія зміни моделюється за допомогою ключового слова when



5.2. Кінцеві автомати. Моделювання поведінки спільно об'єктів

Динамічні аспекти системи в UML моделюються кінцевими автоматами (state machines). Автомат – це поведінка, яка специфікує послідовність станів об'єкта, через які він проходить протягом свого ЖЦ у відповідь на події, а також реакції на ці події

Подія (event) в контексті автоматів - це вплив, який може викликати перехід з одного стану в інший. Під подіями можуть вважатися: вступ сигналу, виклик, закінчення певного періоду часу або зміна стану.

Стан (state) – це ситуація в ЖЦ об'єкта, у якій він задовольняє задані умови, робить якусь діяльність або очікує певної події. Об'єкт перебуває в тому або іншому стані обмежений період часу. Є 4 стани і ми їх розглянемо на прикладі обігрівача у будинку:

1. простий, коли(очікується команда на ввімкнення обігріву),
2. активація, коли (газ поданий, але очікується досягнення певної температури),
3. функціонування, коли (газ і казан ввімкнені),
4. вимикання, коли (подача газу припинилася, але казан увімкнений, залишкове тепло скидається із системи).

5.3. Процеси і потоки керування.

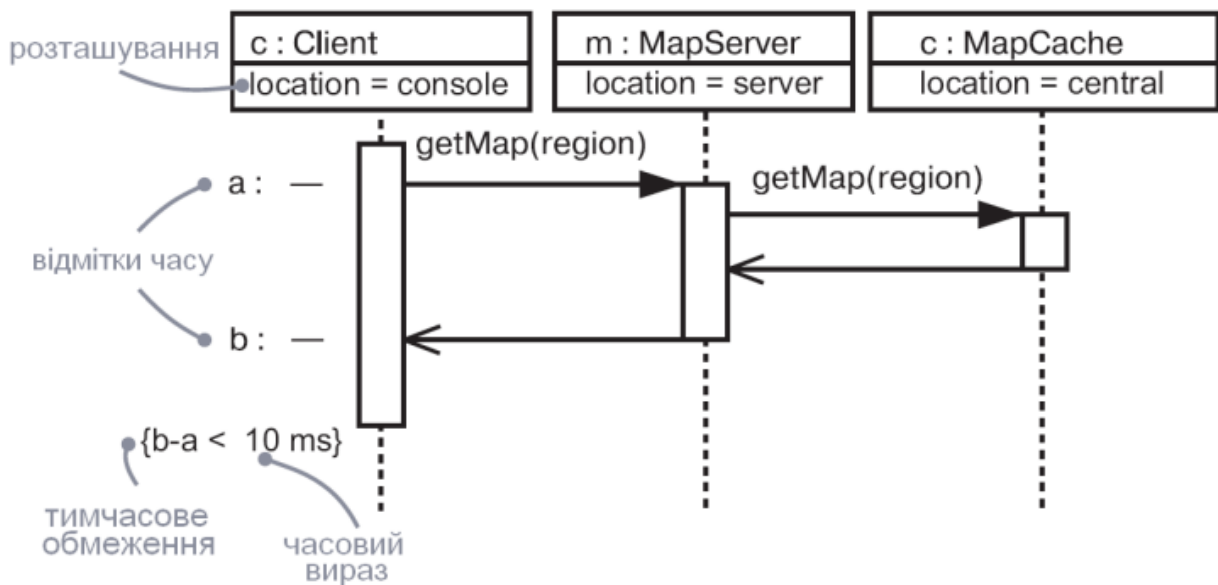
В UML кожен незалежний потік керування моделюється у вигляді активного об'єкта

1. Активний клас (active class) – це клас, екземплярами якого є активні об'єкти.
2. Активний об'єкт (active object) – це об'єкт, який володіє процесом або потоком і може ініціювати керуючий вплив.
3. Процес (process) – це ресурсомісткий потік керування, який виконується паралельно з іншими процесами
4. Потік (thread) – це полегшений потік керування, який може виконуватися паралельно з іншими потоками. Зображуються у вигляді активних класів, а на діаграмах взаємодії виступають у ролі послідовностей.

5.4. Моделювання систем реального часу.

Моделювання часу і простору – важливий елемент будь-якої розподіленої системи або системи реального часу (СРЧ). Якісна модель СРЧ повинна висвітлити всі необхідні й достатні просторово-часові властивості системи.

Система реального часу повинна виконувати свої функції в строго певний абсолютний або відносний момент часу і витратити на це передбачуваний і обмежений час. Серед подібних систем бувають такі, для яких необхідний час реакції обчислюється нано- або мілісекундами, також зустрічаються з часом реакції – секунди і навіть більше.



1. Оцінка часу служить для позначення моменту часу, в який відбулася подія. Вона зображується у вигляді короткої горизонтальної лінії на границі діаграми послідовності

2. Тимчасовий вираз – значенням його є абсолютний або відносний час. Він може бути також сформований з використанням імені повідомлення і вказівки стадії його опрацювання

3. Тимчасове обмеження – це семантичне твердження про відносний або абсолютний час. Графічно зображується – рядком у дужках і пов'язаною з елементом залежністю

4. Місце розташування – розміщення компонента у вузлі. Є атрибутом об'єкта

Діаграми станів.

В UML керована подіями поведінка об'єкта моделюється за допомогою діаграм станів.

Діаграма станів – це просто уявлення автомата, яке підкреслює потік керування від одного стану до іншого.

1. Автомат (state machine) – опис послідовності станів, через які проходить об'єкт протягом життєвого циклу, реагуючи на події, а також опис реакції на ці події.

2. Стан (state) – ситуація в життєвому циклі об'єкта, протягом якої він задовольняє деяку умову, виконує діяльність або очікує якоїсь події

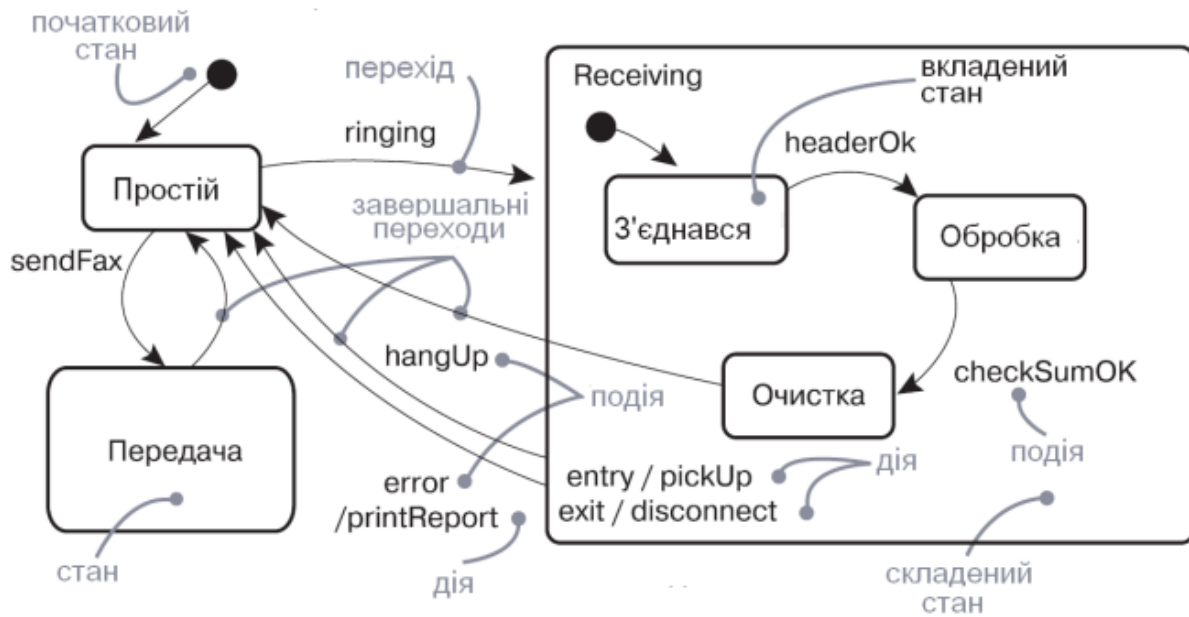
3. Подія (event) – це вплив, який викликає перехід між станами.

4. Перехід (transition) – зв'язок між двома станами, який показує, що об'єкт, який перебуває в першому стані, повинен виконати певні дії і перейти в другий

5. Діяльність (activity) специфікує роботу, що відбувається усередині автомата.

6. Дія (action) – примітивне виконане обчислення, яке призводить до зміни стану моделі або повернення значення

Створюючи діаграму станів в UML, слід пам'ятати, що вона являє собою всього лише проекцію однієї моделі динамічних аспектів системи.



ТЕМА 6

ОСНОВИ МОДЕЛЮВАННЯ АРХІТЕКТУРИ ПЗ

- 6.1. Поняття про архітектуру та раціональний уніфікований процес
- 6.2. Моделювання архітектурних зразків
- 6.3. Моделювання кооперації
- 6.4. Моделювання пакетів як спосіб організації елементів моделі

6.1. Поняття про архітектуру та уніфікований процес.

Архітектура є тим, про що кожен розробник, чи то досвідчений спеціаліст задає собі питання на початковому етапі розробки. Архітектура ПЗ досліджується та вдосконалюється роками і безумовно дуже важливою, але що таке архітектура?

...

Архітектура – це набір суттєвих рішень відносно:

- організації програмної системи;
- вибору структурних елементів, що складають систему, їх інтерфейсів;
- поведінки цих елементів, специфікованої в їхніх коопераціях;
- об'єднання цих структурних елементів та елементів поведінки у більші

підсистеми;

- архітектурного стилю, що визначає організацію системи: статичні й динамічні елементи, їх інтерфейси, кооперацію і композицію.

Якісно зроблена архітектура є фундаментом будь якого професійно розробленого додатку. Вона суттєво зменшує майбутні ризики та витрати. Ігнорувати принципи побудови хорошої архітектури є крайнє необачливо, так вважає відомий програміст та автор багатьох книг Роберт Сесіл Мартін.

Архітектурні вигляди. Архітектура програмної системи може бути описана за допомогою п'яти взаємозалежних архітектурних виглядів Кожен вигляд – проекція організації і структури системи, зосереджена на певному її аспекті.



Вигляд з точки зору **варіантів використання** (Use case view) системи охоплює ВВ, що описують поведінку системи з погляду кінцевих користувачів,

аналітиків і тестувальників. Цей вигляд специфікує не дійсну організацію програмної системи, а ті рушійні сили, що формують системну архітектуру

- Вигляд з точки зору **проектуювання** (Design view) охоплює класи, інтерфейси і кооперації, що формують словник проблеми і її рішення. Цей вигляд, в основному, підтримує функціональні вимоги до системи
- Вигляд з точки зору **процесів** (Process view) системи показує потік керування, що проходить через різні її частини, включаючи можливі механізми паралелізму і синхронізації. Цей вигляд стосується продуктивності, масштабованості й пропускнуої здатності.
- Вигляд з точки зору **реалізації** (Implementation view) системи охоплює артефакти, що використовуються для складання і фізичної реалізації системи. Цей вигляд, у першу чергу, відноситься до керування конфігурацією версій системи.
- Вигляд з точки зору **розгортання** (Deployment view) системи охоплює вузли, що утворюють топологію устаткування, на якому працює система. Цей вигляд, в основному, пов'язаний з розподілом, доставкою та встановленням частин, що складають фізичну систему.

Щоб якомога більше скористатися перевагами процесу розроблення ПЗ, варто розглянути його як процес:

- керований варіантами використання;
- що ґрунтується на архітектурі (архітектурно-керований);
- ітеративний та інкрементний (покроковий).

Ці процеси вже добре відомі нам з курсів інженерії та архітектури ПЗ. Зокрема:

Процес, керований ВВ - поведінка визначається на підставі варіантів використання.

Процес, ґрунтований на архітектурі - архітектура є основним продуктом для концептуалізації і розробки системи.

Ітеративний процес – містить керування потоком виконавчих версій системи.

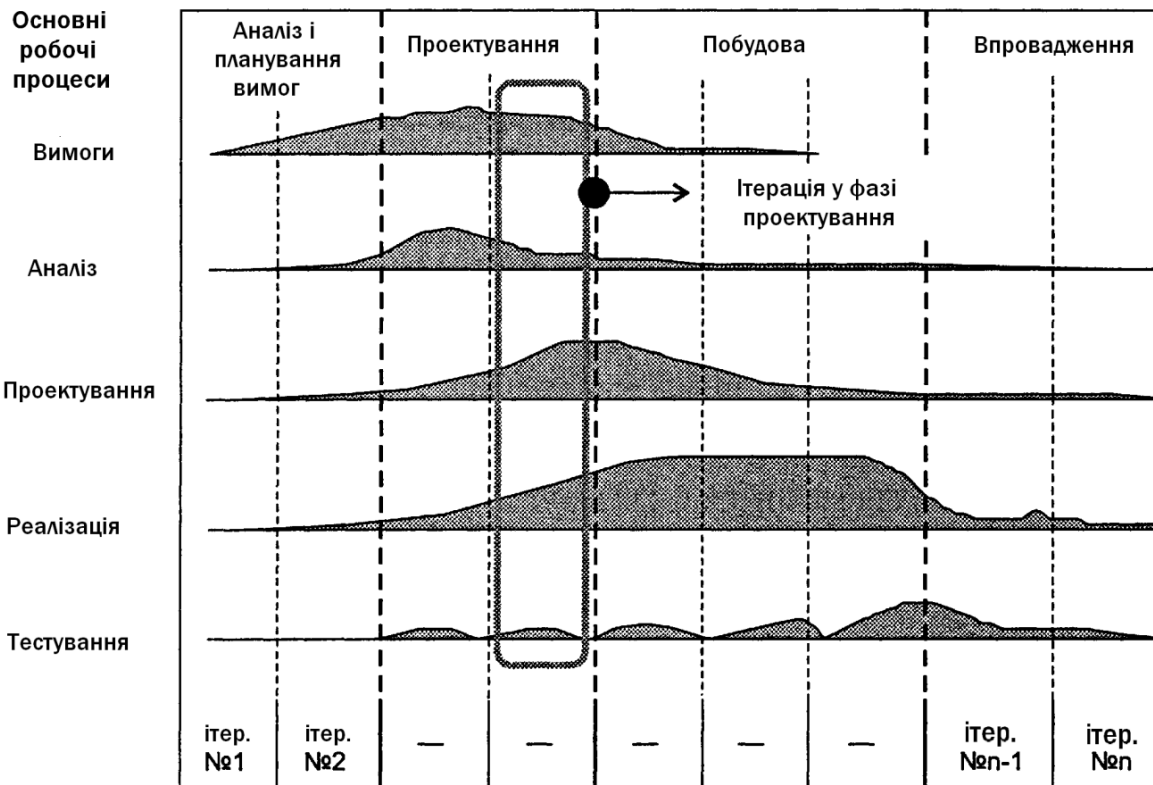
Інкрементний процес ґрунтується на безперервній інтеграції системної архітектури з метою випуску досконаліших версій.

Процес з керованими ризиками - ітеративний і покроковий процес, де при випуску кожної версії серйозна увага приділяється виявленню чинників, що складають найбільший ризик для проекту, та роботі над ними.

Такий керований ВВ процес, сконцентрований на архітектурі, ітеративний і покроковий може бути розбитий на фази.

Фаза – це відрізок часу між двома важливими контрольними точками (milestones) процесу, у яких досягаються добре визначені певні цілі, завершено створення робочих продуктів і ухвалюється рішення про перехід до наступної фази.

Існують чотири фази ЖЦ розроблення ПЗ: аналіз і планування вимог, проектування, побудова і впровадження.



Аналіз і планування вимог (Requirements) – перша фаза процесу, впродовж якої початкова ідея набуває достатнього обґрунтування.

На цій фазі повинні бути отримані відповіді на питання:

- Що повинна робити система, в першу чергу, для її основних користувачів?
- Який вигляд повинна мати архітектура системи?
- Який план і в що обійдеться розроблення продукту?

У **фазі проектування (Design)** детально описуються більшість ВВ і розробляється архітектура системи.

У **фазі побудови (Construction)** відбувається створення продукту – до архітектури (скелету) додаються закінчені програми (м'язи). На цій фазі базовий рівень архітектури розростається до повної розвиненої системи.

Упровадження (Transition) – четверта фаза процесу, у ході якої ПЗ передається користувачам. Але процес розроблення не часто завершується на цьому, тому що навіть на даній фазі система безупинно удосконалюється

Усі добре структуровані системи побудовані на зразках проектування (patterns), що пропонують типові вирішення проблеми в даному контексті.

При моделюванні системи важливим завданням є якісно описати характерні для неї зразки з метою подальшої допомоги команді, яка в майбутньому буде повторно використовувати або модифікувати створюване ПЗ. З практичної точки зору найважливішими є зразки проектування (механізми) і каркаси.

6.2. Моделювання архітектурних зразків

Зразок (pattern) – це загальний розв'язок типової проблеми в даному контексті. Зразки повторюються настільки часто, що їм дають свої імена.

Зразки використовуються для специфікації механізмів і каркасів, що утворюють архітектуру системи.

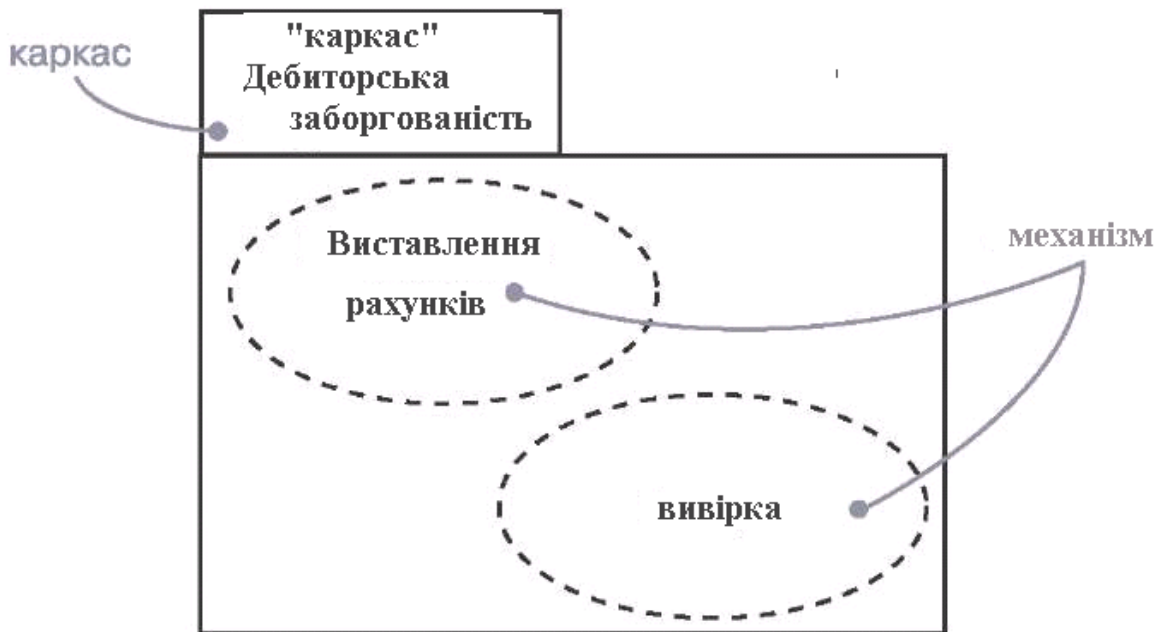
Механізм – це зразок *проектування*, що застосовується до співтовариства класів. Механізм іменує шаблон для набору абстрацій, що працюють спільно для забезпечення деякої типової поведінки (вирішення деякої проблеми)

Каркас (framework) – це *архітектурний* зразок, що пропонує розширюваний шаблон для додатків у деякій предметній області. Каркас можна вважати різновидом мікроархітектури (це більше, ніж механізм).

Каркас містить в собі множину механізмів. При моделюванні каркаса розробляється інфраструктура всієї архітектури, яка потім планується повторно використовуватися і адаптуватися до якогось контексту. Специфікуючи каркас, тим самим описується «кістяк» архітектури.

В UML часто доводиться моделювати зразки проектування (механізми), які можна представити у вигляді кооперацій. Аналогічним чином архітектурні зразки моделюються як каркаси, що представляються у вигляді пакетів зі стереотипами.

Для зразків обох типів в UML передбачений особливий графічний вигляд.



Добре структурована система містить у собі множину зразків на різних рівнях абстракції. Явно виділяючи їх у системі, остання робиться зрозумілішою і простою в супроводі.

6.3 Моделювання кооперації

Кооперація (*collaboration*) – це співтовариство класів, інтерфейсів та інших елементів, які працюють спільно для забезпечення певної поведінки, більш значущої, ніж поведінка суми всіх тих же складових. Кооперація іменує сукупність взаємодіючих будівельних блоків системи, включаючи як структурні, так і поведінкові елементи.

Кожна кооперація повинна мати ім'я, що відрізняє її від інших кооперацій.

Графічне зображення кооперації ви можете побачити на екрані. Крім цього в кооперації виділяють дві її складові: структурну та поведінкову. Структурна складова описується за допомогою таких елементів як класи, інтерфейси, компоненти, які взаємодіють між собою за допомогою різних зв'язків.

Поведінкова складова описується такими артефактами як добре знайомі діаграми взаємодії, послідовності та комунікації.

Існує два види зв'язків, що мають відношення до кооперацій.

Перший: зв'язок між кооперацією і її реалізацією - кооперація може реалізовувати або класифікатор, або операцію. Це означає, що кооперація описує структурну або поведінкову реалізацію відповідного класифікатора чи операції.

Другий: зв'язки між самими коопераціями. Одні з них можуть уточнювати описи інших; це може бути змодельоване у вигляді зв'язку уточнення. Подібні зв'язки між коопераціями відображають, як правило, зв'язки уточнення між ВВ, які вони представляють.

Механізми – це автономні кооперації; їхнім контекстом є не якийсь один варіант використання або одна операція, а система в цілому. Будь-який елемент, видимий у деякій частині системи, – кандидат на участь у механізмі.

Застосування механізмів робить систему простішою (оскільки в механізмі реалізовані типові взаємодії), зрозумілою (тому що до розуміння системи можна підійти з боку її механізмів) і гнучкою (налаштовуючи кожен механізм, налаштовується система в цілому).

Крім кооперації в архітектурному моделюванні також часто використовуються пакети.

6.4. Моделювання пакетів як спосіб організації елементів моделі

Пакет – це спосіб організації елементів моделі в блоки, якими можна розпоряджатися як єдиним цілим. Можна управляти видимістю елементів пакета так, що деякі будуть видні користувачеві, а інші – приховані. Крім того, за допомогою пакетів зображуються різні вигляди архітектури системи. Добреструктурований пакет поєднує семантично близькі елементи, що мають тенденцію змінюватися спільно. Такі пакети характеризуються слабкою зв'язаністю і високою погодженістю, причому доступ до вмісту пакета строго контролюється.

Графічно пакети в UML представлені нотацією у вигляді папки із закладкою

Крім цього, у кожного пакета повинно бути ім'я, що відрізняє його від інших пакетів. Ім'я пакета повинно бути унікальним всередині його пакета

Пакет може володіти іншими елементами, у тому числі класами, інтерфейсами, компонентами, вузлами, коопераціями, ВВ, діаграмами та іншими пакетами.

Пакет визначає свій простір імен, тобто елементи одного виду повинні мати імена, унікальні в контексті пакета, що їх включає.

Видимість належних пакету елементів можна контролювати так само, як видимість атрибутів і операцій класу. За замовчуванням такі елементи є відкритими, тобто видимі для всіх елементів, що втримуються в будь-якому

пакеті, що імпортує даний. Захищені елементи видимі тільки для нащадків, а закриті взагалі невидимі поза своїм пакетом.

Тобто це працює так, як в більшості відомих об'єктно-орієнтованих мовах програмування.

- Order Form (Бланк замовлення) – це відкрита частина пакета Client (Клієнт), а Order (Замовлення) – закрита. Будь-який пакет, що імпортує дані, може «бачити» об'єкт Order Form, але «не бачить» Order.

Найчастіше пакети застосовують для організації елементів моделювання в іменовані групи, з якими потім можна буде працювати як з єдиним цілим.

Між класами і пакетами є одна значна відмінність: класи є абстракції сутностей предметної області, а пакети – це механізми організації таких сутностей у моделі. Це можна представити у вигляді C# неймспейсів, Java пакетів та JS модулів

Використання пакетів для групування родинних елементів досить важливе – без нього не можна розробити складну модель.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Дяченко М.П. Методичні рекомендації щодо забезпечення самостійної роботи студентів з дисципліни «Моделювання та аналіз програмного забезпечення».- К.:МАУП, 2019. -37 с.
2. Кузьмечов А.І. Планування та управління проектами. Моделювання засобами MS Excel. Практикум / Кузьмечов А.І. – Ліга-К.2020.-180.
3. С. Ю. Манаков. Моделювання програмного забезпечення : навч.-метод. посіб. [Електронний ресурс] / уклад.: С. Ю. Манаков, О. Г. Трофименко, Ю. Г. Лобода, А. І. Дика : Нац. ун-т «Одеська юрид. академія». – Одеса : Фенікс, 2023. – 145 с. <http://dspace.onua.edu.ua/handle/11300/13428?locale-attribute=uk>
4. Методичні вказівки і завдання до виконання практичних робіт по курсу «Програмне моделювання динамічних процесів» (для спеціальності 121 Інженерія програмного забезпечення всіх форм навчання) /укл. Дмитрієва О.А.- Дон НТУ, 2020 р. – 102 с.
5. <http://89.185.3.253:9080/download.php?rec=21745>
6. Intech Open. Simulation Modeling. <http://www.intechopen.com>
7. <http://www.ecst.csuchico.edu/~mcleod/software.html>
8. <http://UML.org/>
9. Інформаційні системи і технології: навчальний посібник на інформаційному порталі <http://er.nau.edu.ua>.
10. <http://ukrkniga.org.ua/ukrkniga-text/644/46/>
11. <http://eztuir.ztu.edu.ua/3317/1/17.pdf>
12. Function Points Analysis Training Course. <http://www.softwaremetrics.com>
13. International Function Point Users Group. <http://www.ifpug.org>

Навчально-методичне видання

**Гончар Людмила Іванівна
Марценюк Євгенія Олексіївна**

**ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ
із дисципліни**

**“МОДЕЛЮВАННЯ ТА АНАЛІЗ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ”**

Підписано до друку 05.09.2024 р.
Формат 60x84/16. Папір офсетний.
Друк офсетний. Зам. № 24-759
Умов.-друк. арк. 3,9. Обл.-вид. арк. 4,1.
Тираж 30 прим.

Віддруковано ФО-П Шпак В. Б.
Свідоцтво про державну реєстрацію В02 № 924434 від 11.12.2006 р.
м. Тернопіль, бульвар Просвіти, 6/4. тел. 097 299 38 99.
E-mail: tooums@ukr.net

*Свідоцтво про внесення суб'єкта видавничої справи до державного
реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції
ДК № 7599 від 10.02.2022 р.*