

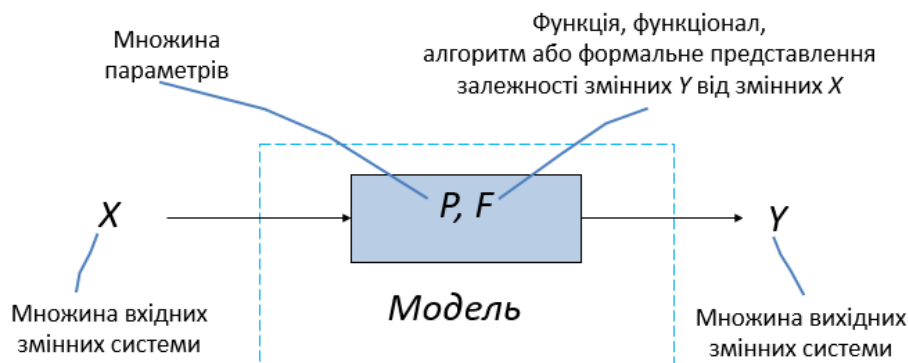
| | | | | | | | |
|----------|---|----|----|----|----|----|----|
| Андрощук | 1 | 7 | 13 | 19 | 25 | 31 | 37 |
| Большой | 2 | 8 | 14 | 20 | 26 | 32 | 38 |
| Максаков | 3 | 9 | 15 | 21 | 27 | 33 | 39 |
| Ноженко | 4 | 10 | 16 | 22 | 28 | 34 | 40 |
| Руденко | 5 | 11 | 17 | 23 | 29 | 35 | 41 |
| Янголь | 6 | 12 | 18 | 24 | 30 | 36 | 42 |

1. Поняття моделі. Способи побудови моделей. Класифікація моделей.

• Моделі – основний засіб наукового дослідження об'єктів і систем

Моделлю називається представлення об'єкта, системи чи поняття в деякій абстрактній формі, що є зручною для наукового дослідження.

Загальна структура моделі



- Модель завжди є спрощенням реального процесу/ системи.
- Від рівня деталізації опису процесів залежить складність моделі.
- Мистецтво дослідника, який будує модель, полягає в тому, щоб для найменш складної моделі отримати найбільш точні результати для задачі, яка поставлена

Способи побудови моделей

Перший спосіб: в результаті ретельного вивчення системи встановлюються закони функціонування системи, які потім відтворюються за допомогою моделі => **Фізичні** моделі

Другий спосіб: без усякого фізичного обґрунтування припускається вид залежності F , невідомі параметри якої P потім відшукуються за даними спостережень за змінними системи X, Y => **Нефізичні** моделі

Системний підхід до побудови моделей

Системний підхід до дослідження систем означає, що дослідник вивчає функціонування системи в цілому, не концентруючи свою увагу на окремих її частинах.

Оснований системний підхід на визнанні факту, що навіть найліпше функціонування окремих підсистем та елементів системи не гарантує найліпшого функціонування всієї системи в цілому, оскільки завжди існує взаємодія між частинами системи.

Опис системи разом із указуванням цілі та задачі дослідження складає концептуальну модель системи.

Розробка концептуальної моделі системи складає основну задачу системного аналізу об'єкта, явища чи поняття, що досліджується.

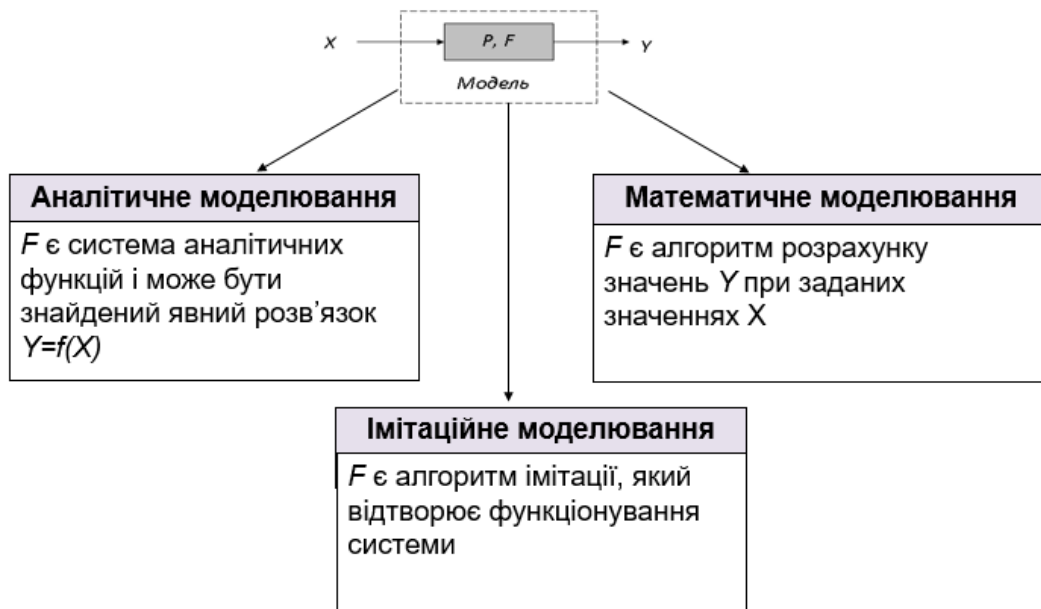
Класифікація моделей

- 3 точки зору вихідної змінної моделі:
 - ❖ **статичні та динамічні**
 - **неперервні та дискретні**
 - **детерміновані та стохастичні**
- 3 точки зору способу побудови моделі:
 - ❖ **фізичні та нефізичні**
- 3 точки зору методу моделювання:
 - ❖ **алгебраїчні, диференційні, аналітичні, імітаційні,...**

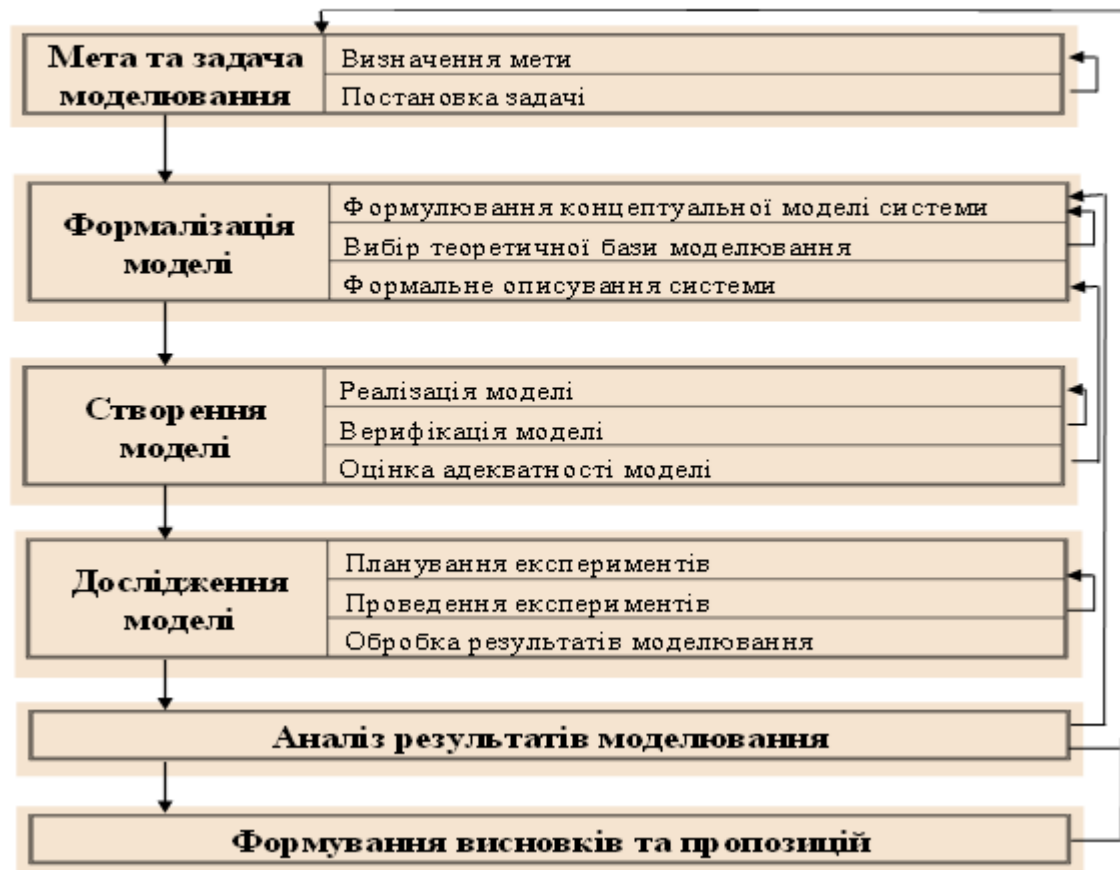
•Стохастичні моделі - ті, що використовують для опису своїх змінних та/або параметрів випадкові величини

2. Методи моделювання. Технології моделювання.

Методи моделювання



Процес моделювання



Програмне забезпечення моделювання систем

Неперервні моделі: Matlab, Simulink

Дискретно-подійні моделі: VisSim, CPNTools, AnyLogic,
ProModel, Simio <https://www.simio.com/index.php>

Arena Simulation Software <https://www.arenasimulation.com/>

3. Генератори випадкових чисел.

Випадкова величина - спрощене представлення складних процесів, які впливають на значення величини.

- зберігання у комп'ютері таблиці випадкових чисел і отримання потім з неї даних для імітаційного моделювання;
- використання деякого фізичного пристрою (електронна лампа для генерації випадкового шуму);
- застосування рекурсивних формул коли на підставі i -того випадкового числа обчислюється $i+1$ -ше випадкове число.

Генератори випадкових величин:

- генерування рівномірно розподіленої в інтервалі $(0;1)$ випадкової величини ζ
- генерування випадкової величини γ за заданим законом розподілу $F(x)$
- метод оберненої функції (експоненціальний закон розподілу)
- табличний метод (емпіричний закон розподілу)
- спеціальні методи (закон Гауса(норм закон розподілу), закон розподілу Ерланга

4. Способи генерування рівномірно розподіленої в інтервалі $(0;1)$ випадкової величини.

Генерування рівномірно розподілених
в інтервалі (0;1) випадкових величин
на основі рекурсивних формул

$$z_{i+1} = (az_i + b)(\text{mod } c), i = 0, 1, \dots$$

$$\zeta_{i+1} = z_{i+1} / c$$

Тестування генераторів рівномірно розподілених в інтервалі (0,1) випадкових чисел:

- перевірка на рівномірність,
- перевірка на випадковість,
- кореляції

5. Формалізація процесів функціонування дискретно-подійних систем.

Найбільш формалізованим представленням є представлення у вигляді математичних формул. Менш формалізованим, але більш універсальним – представлення у вигляді схем, що відображають елементи системи та структурні взаємозв'язки між ними. Найбільш відомими і найбільш поширеними серед спеціалістів засобами формалізації процесів функціонування дискретних систем являються мережі масового обслуговування та мережі Петрі. Мережі масового обслуговування призначені для описування процесів обробки, таких як виробничі процеси, бізнес-процеси, і представляють великий, але чітко обмежений клас систем. Мережі Петрі являються більш могутнім засобом формалізації дискретних процесів і описують системи із найскладнішими зв'язками, в тому числі управляючими.

| | | |
|---|---------------------------------------|---|
| Формалізація процесів функціонування дискретних систем | Мережі масового обслуговування | <ul style="list-style-type: none"> ▪ розімкнуті ▪ замкнуті ▪ з блокуванням маршруту |
| | Мережі Петрі | <ul style="list-style-type: none"> ▪ з часовими затримками ▪ з конфліктними переходами ▪ з багатоканальними переходами ▪ з інформаційними зв'язками |

6. Процес формалізації дискретно-подійної системи.

хз-хз(Старт

- Ініціалізувати Умову Завершення зі значенням FALSE.
- Ініціалізувати змінні стану системи.
- Ініціалізувати Годинник (зазвичай починається з нульового часу).
- Запланувати початкову подію (тобто внести якусь початкову подію до Списку Подій).

Цикл виконання

Поки (Умову Завершення має значення FALSE), виконувати наступне:

- Встановити Годинник на час наступної події.
- Зробіть наступну подію та видалити зі Списку Подій.
- Оновити Статистику.

Кінець

- Створити статистичний звіт.)

7. Формалізм мереж масового обслуговування.

Послідовність дій, виконуваних для формалізації системи засобами мережі масового обслуговування

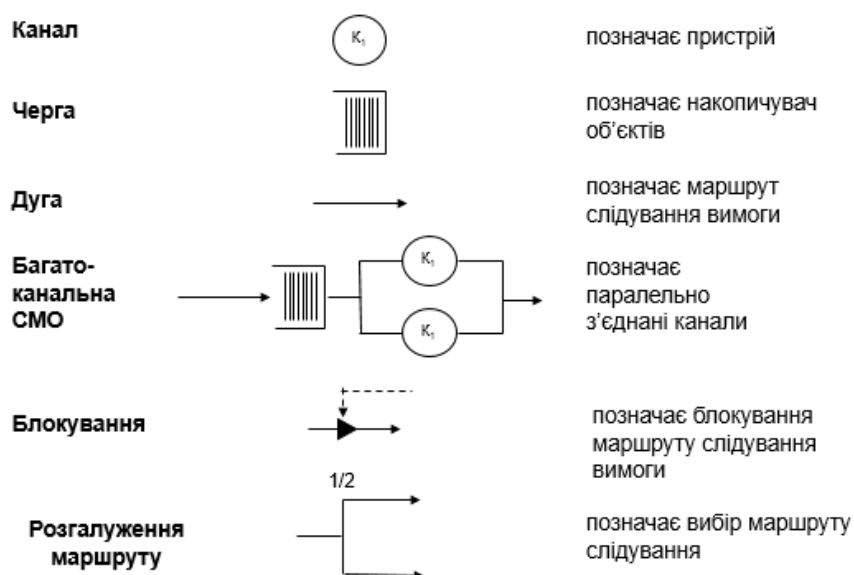
Для того, щоб представити систему мережею масового обслуговування потрібно:

- з'ясувати, що в системі є об'єктом обслуговування;
- виділити елементи процесу обслуговування об'єктів і кожному елементу поставити у відповідність СМО;
- для кожної СМО визначити кількість пристроїв та наявність черги;
- з'єднати СМО у відповідності до процесу обслуговування;
- визначити маршрут проходження об'єкту обслуговування від однієї СМО до іншої;
- визначити умови надходження в кожен СМО (ймовірність вибору маршруту та інші);
- визначити наявність блокування маршруту та умови блокування;
- визначити числові значення параметрів кожної СМО;
- визначити числові значення параметрів зовнішнього потоку на обслуговування;
- визначити стан мережі масового обслуговування на початку моделювання.

Параметри та вхідні змінні мережі МО

- Кількість СМО.
 - Для кожної СМО:
 - кількість пристроїв,
 - обмеження на чергу (або його відсутність),
 - параметри часу обслуговування.
 - Для замкнутої мережі МО - кількість замовлень, які циркулюють в мережі
 - Для розімкнутої мережі МО – параметри інтервалу надходження нового замовлення
 - Для маршруту слідування
 - ймовірності слідування з однієї СМО в іншу,
 - для розімкнутої мережі, ймовірності надходження замовлення ззовні до однієї зі СМО,
 - наявність блокування маршруту між СМО та умова блокування
- Ймовірність відмови
 - Середнє завантаження пристроїв
 - Середня довжина черги
 - Середнє очікування в черзі

Елементи мережі масового обслуговування



8. Алгоритм імітаційного моделювання дискретно-подійної системи.

Дискретно-подійна система (discrete-event system) = система, яка описується дискретною множиною станів і множиною подій (discrete-state, event-driven system). Функціонування системи повністю визначається послідовністю (асинхронних) подій, які відбуваються в часі. Алгоритм, який відтворює функціонування системи, за допомогою комп'ютерної програми називається алгоритмом імітації.

Алгоритм імітації дискретно-подійної системи відтворює упорядковану в часі послідовність подій

Узагальнена схема алгоритму імітації



Способи просування модельного часу

- ✓ за принципом Δt ,
- ✓ за принципом найближчої події,
- ✓ за принципом послідовного проведення об'єктів уздовж моделі.

Спосіб, орієнтований на події

При підході, орієнтованому на події, дослідник визначає і описує події, які виникають у моделі. Імітація здійснюється виконанням упорядкованої у часі послідовності логічно взаємозв'язаних подій.

Проілюструємо на прикладі моделі обслуговування одним пристроєм потоку вимог (див. рисунок 4.1). Стан системи описується станом пристрою обслуговування та станом черги. Змінювання стану системи відбувається у моменти, коли у систему надходить вимога і коли закінчився процес обслуговування вимоги у пристрої. Назвемо ці події подія «надходження» та подія «закінчилось обслуговування у пристрої».

Подія «надходження» складається з таких дій:

- якщо пристрій обслуговування в стані вільний установити пристрій у стан «зайнятий», запам'ятати момент виходу вимоги з пристрою в момент часу - поточний момент часу плюс тривалість обслуговування у пристрої;
- інакше
 - якщо є вільне місце в черзі, то зайняти місце в черзі;
 - інакше збільшити кількість не обслугованих вимог на одиницю;
- генерувати момент надходження наступної вимоги у СМО.

Подія «закінчилось обслуговування в пристрої» складається з таких дій:

- збільшити кількість обслугованих вимог на одиницю;
- якщо черга вимог не пуста, перемістити одну вимогу із черги у канал обслуговування, запам'ятати момент виходу вимоги з пристрою у момент часу - поточний момент часу плюс тривалість обслуговування у пристрої;
- інакше установити канал у стан «вільний» вільний стан, запам'ятати момент виходу вимоги з пристрою у момент часу, що більший за час моделювання (тобто у найближчий час вихід вимоги із пристрою не очікується).

Після виконання події «надходження» запам'ятовуються моменти виникнення наступних подій «надходження» та «закінчилось обслуговування у пристрої». З цих двох моментів часу вибирається найменший і запам'ятовується, яка саме подія відповідає цьому моменту. Модельний час просувається у визначений момент найближчої події і виконується відповідна йому подія.

При виконанні події «закінчилось обслуговування у пристрої» запам'ятовується момент наступної події «закінчилось обслуговування у пристрої» або запам'ятовується, що найближчим часом не очікується звільнення пристрою.

Таким чином, події виконуються в упорядкованій у часі послідовності, а модельний час просувається від одного моменту виникнення події до найближчого наступного.

Програма, яка здійснює імітацію моделі, що розглядається, складається з таких дій:

- 1) ввести початкові значення змінних - поточний час, час моделювання, стан пристрою, момент надходження вимоги у систему, момент звільнення пристрою обслуговування;
- 2) доки поточний час менший за час моделювання
- 3) знайти найменший із моментів часу «момент надходження вимоги у систему» та «момент звільнення пристрою обслуговування» і запам'ятати, якій події він відповідає;
- 4) просунути поточний час у момент найближчої події;
- 5) виконати подію, яка відповідає моменту найближчої події.
- 6) вивести результати моделювання – кількість обслугованих вимог та кількість не обслугованих вимог протягом часу моделювання.

9. Алгоритм імітації на основі подійного представлення процесу функціонування та просування часу до найближчої події.

Принцип найближчої події

Дискретні системи, імітаційне моделювання яких розглядається, мають певну особливість: змінювання стану в таких системах відбувається тільки в деякі моменти часу, а в усі інші моменти часу система не змінюється. Змінювання стану моделі спричиняється виникненням певної події у системі. Наприклад, подія «надходження деталі до технологічного процесу» спричиняє збільшення кількості деталей у системі, подія «виникнення поломки обладнання» спричиняє відправку пошкодженого обладнання до ремонту і т.д. Процес функціонування системи розглядається як послідовність подій, що відбуваються у моделі.

За принципом найближчої події модельний час просувається від моменту виникнення однієї події до моменту виникнення іншої, і після кожного просування часу реалізуються зміни стану моделі, відповідні до події, що виникла. Використання принципу найближчої події вимагає від дослідника побудови спеціальної процедури визначення моменту найближчої події, але при цьому він отримує вигоду у затратах комп'ютерного часу, оскільки пропускається моделювання системи у моменти часу, коли події не відбуваються.

Розглянемо, наприклад, найпростішу модель обслуговування одним пристроєм потоку вимог (рисунок 2.3). В системі спостерігаються дві події: V - надходження вимоги, W - звільнення пристрою обслуговування. Представимо моменти часу, коли спостерігалась подія V , та моменти часу, коли спостерігалась подія W .

Упорядкуємо моменти виконання подій V і W на вісі модельного часу t_m . На першому кроці алгоритму імітації встановлюється момент модельного часу t_{m1} та реалізується подія V ; на другому - момент модельного часу t_{m2} та реалізується подія W .

W_1 , на третьому - t_{m3} та реалізуються події V_2 і W_2 у відповідності з пріоритетом процесів, і т.д. Для дискретних систем просування модельного часу за принципом найближчої події є найефективнішим способом.

10. Об'єктно-орієнтований підхід до розробки алгоритму імітаційного моделювання дискретно-подійних системи.(лекція 4 одні блять класи да свойства с методами)

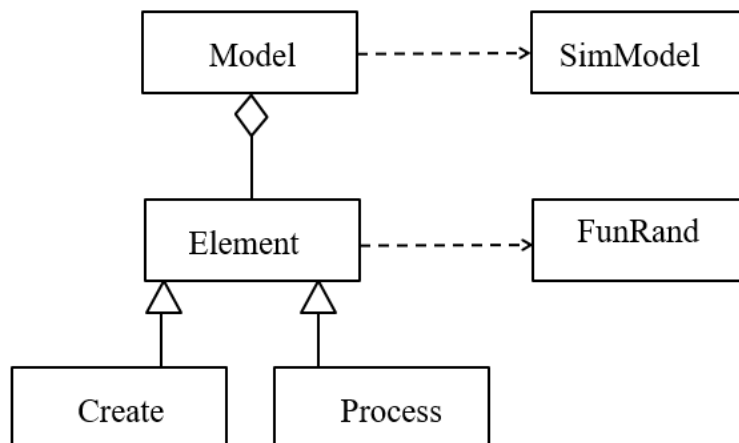
Розглянемо побудову алгоритму імітації для простої моделі масового обслуговування, представленої на рисунку 2.2. Вимоги на обслуговування генеруються елементом CREATE і відправляються на обслуговування до елементу PROCESS, який здійснює обслуговування з часовою затримкою, заданою випадковим числом, та обмеження на довжину черги, заданим невід'ємним числом.

Структуру об'єктно-орієнтованої програми представимо діаграмою класів (рис. 2.3). Модель складається з елементів, які є нащадками одного універсального типу Element. Цей клас містить основні поля та методи елемента моделі такі, як `tcurr` (поточний момент часу), `tnext` (момент часу наступної події), `delayMean` (середнє значення часової затримки), `delayDev` (середнє квадратичне відхилення часової затримки), `getDelay()` (розрахунок часової затримки), `inAct()` (вхід в елемент), `outAct()` (вихід з елемента). Використання універсального класу надає можливість уніфікувати використання різних елементів в імітаційній програмі.

Важливим для з'єднання елементів в єдину модель є поле `nextElement`, що вказує на наступний (в маршруті слідування вимоги) елемент моделі.

Клас Model містить метод `simulate(double time)`, що здійснює імітацію на інтервалі часу `time`. Імітація здійснюється за відомим з попередньої теми принципом: визначається момент найближчої події, просувається час в момент найближчої події та здійснюється відповідна подія. Щоб зменшити обсяг обчислень, введемо також здійснення відповідної події для всіх елементів, час наступної події яких співпадає з поточним моментом часу. Оскільки ми не розглядаємо можливість присвоєння пріоритету елементам моделі, то ця дія цілком допустима.

Діаграма класів об'єктно-орієнтованої програми імітації дискретно-подійної системи



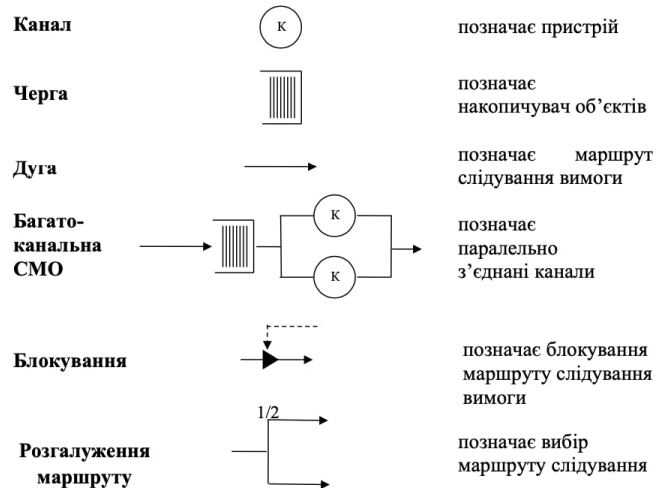
11. Універсальний алгоритм імітації мережі масового обслуговування.

Алгоритм імітації мережі масового обслуговування складається з опису елементів моделі, опису стану елементів моделі та опису множини подій.

Елементами мережі масового обслуговування являються:

- вхідний потік,
- система масового обслуговування,
- зв'язок.

ЕЛЕМЕНТИ МЕРЕЖІ МАСОВОГО ОБСЛУГОВУВАННЯ



Кожний пристрій системи масового обслуговування характеризується тривалістю обслуговування вимоги та станом пристрою.

Множина подій мережі масового обслуговування складається з подій двох типів:

- надходження вимоги в мережу масового обслуговування;
- вихід вимоги, для якої закінчився час обслуговування, з системи масового обслуговування.

Використовуємо способи:

- просування часу за принципом до найближчої події,
- орієнтований на події спосіб просування моделі в часі.

Об'єктно-орієнтований підхід

Розгалуження маршруту:

- за заданою ймовірністю
- за пріоритетом

Багатоканальність обслуговування

Блокування маршрутів за умовою, що враховує стан мережі або окремих її елементів

Емпіричний закон розподілу

Узагальнена схема алгоритму імітації



12. Верифікація алгоритму імітації.

Верифікація алгоритму імітації

Верифікація алгоритму імітація = перевірка відповідності задуму моделювання.

При змінюванні вхідної змінної моделі перевіряємо, як ця зміна впливає на вихідні змінні. Робимо висновок про правильність чи неправильність функціонування моделі.

Верифікація моделі масового обслуговування

| Прогон | Значення вхідних змінних | | | | | | | | | | | Значення вихідних змінних моделі | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------|-------------------------------------|---|----------------------------|-----|---------------------------------|---|--|---|----------------------------|---|---------------------------------|----------------------------------|--|----|----------------------------|------|---------------------------------|-----|--|-----|-----------------------------|-----|---------------------------------------|--|----------------------------------|--|----------------------------------|--|----------------------------------|--|---------------------|--|------------------------------|--|---|--|------------------------------|--|---|--|------------------------------|--|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| | Середній інтервал надходження вимог | | Кількість пристроїв в СМО1 | | Обмеження на довжину черги СМО1 | | Середня тривалість обслуговування в СМО1 | | Кількість пристроїв в СМО2 | | Обмеження на довжину черги СМО2 | | Середня тривалість обслуговування в СМО2 | | Кількість пристроїв в СМО3 | | Обмеження на довжину черги СМО3 | | Середня тривалість обслуговування в СМО3 | | Ймовірність вибору маршруту | | Кількість вимог, що надійшли в мережу | | Кількість не обслугованих в СМО1 | | Кількість не обслугованих в СМО2 | | Кількість не обслугованих в СМО3 | | Ймовірність відмови | | Середня довжина черги в СМО1 | | Середня кількість зайнятих пристроїв СМО1 | | Середня довжина черги в СМО2 | | Середня кількість зайнятих пристроїв СМО2 | | Середня довжина черги в СМО3 | | Середня кількість зайнятих пристроїв СМО3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0,2 | 5 | 10 | 1,2 | 7 | 8 | 2 | 2 | 1 | 1 | 0,7 | 4863 | 872 | 21 | 130 | 0,20 | 6,3 | 4,9 | 1,1 | 5,5 | 0,10 | 1,0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Верифікація моделі іде після реалізації моделі. Після верифікації моделі іде оцінка адекватності моделі.

13. Оцінка точності алгоритму імітації.

Оцінка точності алгоритму імітації мережі масового обслуговування

Виконується порівнянням з результатами аналітичного розрахунку.

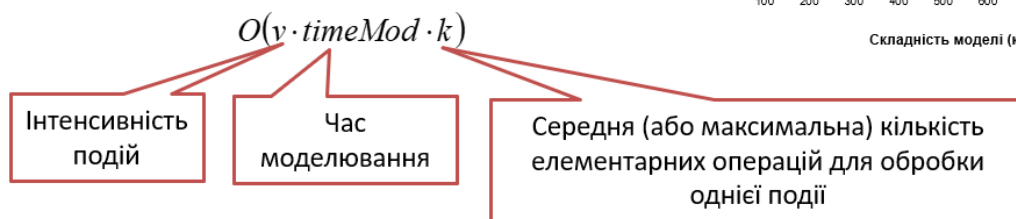
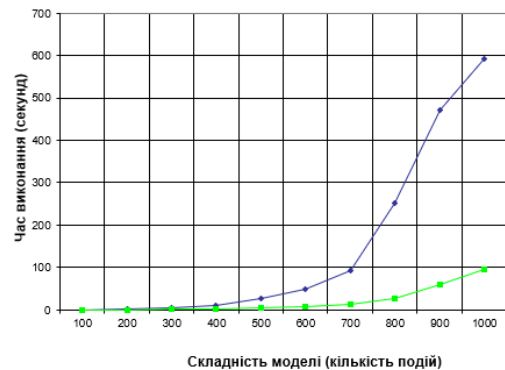
Аналітичний розрахунок можливий за таких умов:

- усі черги необмеженої довжини
- усі часові затримки задані випадковими величинами з експоненціальним законом розподілу
- вибір маршруту виключно за заданими ймовірностями
- блокування маршрутів відсутні

14. Оцінка складності алгоритму імітації.

Оцінка складності алгоритму імітації для мережі масового обслуговування

- Експериментальна оцінка складності виконується побудовою залежності часу виконання алгоритму в залежності від складності моделі
- Теоретична оцінка складності виконується підрахунком кількості елементарних операцій в алгоритмі в залежності від складності моделі:



15. Формалізм базових мереж Петрі.

16. Формалізм стохастичних мереж Петрі.

Переваги формалізації мережею Петрі:

Гнучкість формалізації подій

Універсальність алгоритму імітації

Візуалізація процесу функціонування

Пристосованість до представлення паралельних процесів

Недоліки формалізації мережі Петрі:

Велика кількість елементів

Елементи мережі Петрі

ЕЛЕМЕНТИ МЕРЕЖІ ПЕТРІ



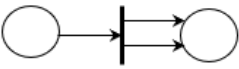


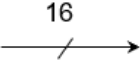
| | | |
|------------------------|---|---|
| Перехід |  | позначає подію |
| Позиція |  | позначає умову |
| Дуга |  | позначає зв'язки між подіями та умовами |
| Маркер(один) (один) |  | позначає виконання (або не виконання) умови |
| Багато фішок |  | позначає багатократне виконання умови |
| Багато дуг |  | позначає велику кількість зв'язків |

Рисунок 3.6. Елементи мережі Петрі

Формальне означення стохастичної мережі Петрі

$$N = (P, T, A, W, K, R)$$

$P = \{P\}$ - множина позицій;

$T = \{T\}$ - множина переходів;

$$P \cap T = \emptyset$$

$A \subseteq (P \times T \cup T \times P)$ - множина дуг;

$W: A \rightarrow \mathbb{N}$ - множина натуральних чисел, що задають кратності дуг (кількість зв'язків);

$K = \{(c_T, b_T) | T \in T, c_T \in \mathbb{N}, b_T \in [0; 1]\}$ - множина пар значень, що задають пріоритет та ймовірність запуску переходів;

$R: T \rightarrow \mathcal{R}_+$ - множина невід'ємних чисел, що характеризують часові затримки

Формальне означення стохастичної мережі Петрі з інформаційними зв'язками

$$N = (P, T, A, W, I, K, R)$$

$P = \{P\}$ - множина позицій;

$T = \{T\}$ - множина переходів;

$$P \cap T = \emptyset$$

$A \subseteq (P \times T \cup T \times P)$ - множина дуг;

$I \subseteq (P \times T)$ - множина інформаційних дуг;

$W: A \cup I \rightarrow \mathbb{N}$ - множина натуральних чисел, що задають кратності дуг (кількість зв'язків);

$K = \{(c_T, b_T) | T \in T, c_T \in \mathbb{N}, b_T \in [0; 1]\}$ - множина пар значень, що задають пріоритет та ймовірність запуску переходів;

$R: T \rightarrow \mathcal{R}_+$ - множина невід'ємних чисел, що характеризують часові затримки

17. Універсальний алгоритм імітації стохастичної мережі Петрі.

Алгоритм імітації мережі Петрі складається з опису елементів моделі та опису правил зміни стану елементів.

Елементами мережі Петрі являються:

- позиція,
- перехід,
- зв'язок.

Стан позиції повністю описується кількістю маркерів у позиції.

Алгоритм імітації стохастичної мережі Петрі відрізняється від алгоритму імітації мережі Петрі тим, що затримки в стохастичний визначаються випадковим чином.

$K = \{(c_T, b_T) | T \in T, c_T \in \mathbb{N}, b_T \in [0; 1]\}$ - множина пар значень, що задають пріоритет та ймовірність запуску переходів;

$R: T \rightarrow \mathcal{R}_+$ - множина невід'ємних чисел, що характеризують часові затримки

Доки $t < T_{\text{mod}}$

визначити момент найближчої події \min ;

зібрати статистику про функціонування моделі; $t = \min$;

якщо $t < T_{\text{mod}}$

виконати вихід маркерів з переходів мережі Петрі:

виконати вихід маркерів з переходу, що відповідає моменту найближчої події:

збільшити кількість маркерів в позиції на відповідне число

та запам'ятати момент виходу з переходу як рівний «нескінченності»

для кожного переходу:

якщо момент виходу маркерів з переходу співпадає з поточним часом,

виконати вихід маркерів з цього переходу:

збільшити кількість маркерів в позиції на відповідне число

та запам'ятати момент виходу з переходу як рівний «нескінченності» виконати вхід маркерів в переходи мережі Петрі:

визначити список переходів з виконаною умовою запуску та вибрати з них один
(за заданими значеннями пріоритету та ймовірності запуску)

доки список переходів з виконаною умовою запуску непорожній виконати вхід маркерів в перехід:

зменшити кількість маркерів у відповідних позиціях та запам'ятати нове значення
моменту виходу маркерів з переходу;

визначити список переходів з виконаною умовою запуску та вибрати з них один
(за заданими значеннями пріоритету та ймовірності запуску);

Доки $t < T_{mod}$

визначити момент найближчої події t_{in} ;

зібрати статистику про функціонування моделі;

$t = t_{in}$;

якщо $t < T_{mod}$

виконати вихід маркерів з переходів мережі Петрі:

виконати вихід маркерів з переходу, що відповідає моменту найближчої події:

збільшити кількість маркерів в позиції на відповідне число

та запам'ятати момент виходу з переходу як рівний «нескінченності»

для кожного переходу:

якщо момент виходу маркерів з переходу співпадає з поточним часом,

виконати вихід маркерів з цього переходу:

збільшити кількість маркерів в позиції на відповідне число

та запам'ятати момент виходу з переходу як рівний «нескінченності»

виконати вхід маркерів в переходи мережі Петрі:

визначити список переходів з виконаною умовою запуску та вибрати з них один
(за заданими значеннями пріоритету та ймовірності запуску)

доки список переходів з виконаною умовою запуску непорожній

виконати вхід маркерів в перехід:

зменшити кількість маркерів у відповідних позиціях та запам'ятати нове значення
моменту виходу маркерів з переходу;

визначити список переходів з виконаною умовою запуску та вибрати з них один
(за заданими значеннями пріоритету та ймовірності запуску);

18. Алгоритм імітації стохастичної мережі Петрі з багатоканальними переходами.

Багатоканальними називають переходи з часовою затримкою, в яких одночасно може знаходитись декілька маркерів.

Алгоритм імітації мережі Петрі з багатоканальними переходами ускладнюється, по-перше, тим, що замість одного значення t_outj у переході, що є багатоканальним, доведеться запам'ятовувати масив значень t_outj . По-друге, доведеться передбачити багаторазовий запуск переходу для чого перевірку умови запуску переходу та запуск переходів потрібно здійснювати доти, доки виконується умова запуску хоч для одного з них.

- для кожного переходу T_j :
 - доки $k=0$ виконувати багатократний запуск переходу T_j :
 - $k:=0$;

19. Алгоритм імітації мережі Петрі з часовими затримками, з багатоканальними переходами, з конфліктними переходами.

Якщо умова запуску переходу виконана, здійснюється вхід маркерів до переходу і запам'ятовується момент виходу маркерів з переходу. Потім обирається перехід, для якого момент виходу маркерів з переходу найменший, і здійснюється вихід маркерів з обраного переходу. Вихід маркерів з переходу змінює маркірування мережі Петрі, тому знову перевіряється умова запуску переходу і т.д. Процес продовжується, доки запускається хоч один перехід, або, доки вичерпаний час спостереження мережі Петрі.

20. Математичний опис стохастичної мережі Петрі.

Мережа Петрі:

$$N = (\mathbf{P}_N, \mathbf{T}_N, \mathbf{A}_N, \mathbf{W}_N, \mathbf{K}_N, \mathbf{I}_N, \mathbf{R}_N)$$

$\mathbf{P}_N = \{P\}$ - множина позицій

$\mathbf{T}_N = \{T\}$ - множина переходів $\mathbf{P}_N \cap \mathbf{T}_N = \emptyset$

$\mathbf{A}_N \subseteq (\mathbf{P}_N \times \mathbf{T}_N \cup \mathbf{T}_N \times \mathbf{P}_N)$ - множина дуг

$\mathbf{W}: \mathbf{A} \cup \mathbf{I} \rightarrow \mathbf{N}$ - кратності дуг

$\mathbf{K} = \{(c_T, b_T) \mid T \in \mathbf{T}, c_T \in \mathbf{N}, b_T \in [0;1]\}$ - пріоритети та ймовірності запуску переходів

$\mathbf{R}: \mathbf{T} \rightarrow \mathfrak{R}_+$ - часові затримки в переходах

$\cdot T, T \cdot$ - множина вхідних та множина вихідних позицій переходу T

$\cdot P, P \cdot$ - множина вхідних і множина вихідних переходів позиції P

$\mathbf{S}(t) = (\mathbf{M}(t), \mathbf{E}(t))$ - стан стохастичної мережі Петрі в момент часу t

$\mathbf{M}(t) = \{M_P(t) \mid M_P(t) \in \mathbf{Z}_+, P \in \mathbf{P}\}$ - стан позицій

$\mathbf{E}(t) = \{E_T(t) \mid T \in \mathbf{T}\}$ - стан переходів

$E_T(t) = \{ [E_T(t)]_q \mid [E_T(t)]_q \in \mathfrak{R}_+, q \in \mathbf{N} \}$ - стан переходу T ,
 q – номер запланованої події виходу маркерів з переходу в момент часу t

$E_T(t) = \{\infty\}$ якщо найближчим часом не очікується вихід маркерів з переходу

$\mathbf{S}(t) \in \{ \mathbf{S}(t) \mid (M_P(t) \geq 0 \forall P \in \mathbf{P}) \wedge ([E_T(t)]_q \geq 0 \forall T \in \mathbf{T}, \forall q = 1, \dots, |E_T(t)|) \}$

Визначення моменту найближчої події

$\tau_T(t) = \min_q [E_T(t)]_q$, - момент запланованої найближчої події для переходу T на поточний момент часу

$t' = \min_T \tau_T(t), t' \geq t$ - момент запланованої найближчої події для мережі Петрі на поточний момент часу

$$t_n = \min_T \left(\min_q [E_T(t_{n-1})]_q \right), t_n \geq t_{n-1}$$

- визначення моменту найближчої події для мережі Петрі на поточний момент часу

Вихід маркерів з переходів

$Y : T \times \mathbb{R} \rightarrow \{0;1\}$ - предикат, що визначає для кожного переходу співпадіння моменту найближчої події з поточним моментом часу

$$(\tau_T(t) = t') \Rightarrow Y(T, t') = 1,$$

$$(\tau_T(t) \neq t') \Rightarrow Y(T, t') = 0.$$

$$\forall P \in \mathbf{P} \quad M_P^+(t') = M_P(t) + \sum_{T \in {}^*P} Y(T, t') \cdot W_{T,P} |s_T(t)|, \quad - \text{змінювання стану позиції } P$$

$$\forall T \in \mathbf{T} \mid Y(T, t') = 1 \quad E_T^+(t') = \begin{cases} \{\infty\} \leftarrow |s_T(t)| = |E_T(t)| \\ E_T^+(t') = E_T(t) \setminus \{[E_T(t)]_q \mid q \in s_T(t)\} \leftarrow |s_T(t)| \neq |E_T(t)| \end{cases}$$

- змінювання стану переходу T

$$D^+ : \mathbf{E}(t) \times \mathbf{M}(t) \rightarrow \mathbf{E}(t') \times \mathbf{M}(t') \quad - \text{перетворення стану стохастичної мережі Петрі}$$

$$\mathbf{S}(t') = D^+(\mathbf{S}(t)) \quad - \text{змінювання стану стохастичної мережі Петрі}$$

Вхід маркерів в переходи

$Z : T \times \mathbb{R} \rightarrow \{0;1\}$ - предикат, що визначає для кожного переходу умову виконання запуску

$$(\forall P \in {}^*T \quad M_P^+(t') \geq W_{P,T}) \Rightarrow Z(T, t') = 1$$

$$(\exists P \in {}^*T \quad M_P^+(t') < W_{P,T}) \Rightarrow Z(T, t') = 0$$

$$\Psi(t') = \{T \mid Z(T, t') = 1, T \in \mathbf{T}\} \quad - \text{множина переходів, для яких виконана умова запуску}$$

$\Psi'(t') = \overline{\mathbf{T}_\Psi} \cup \widetilde{\mathbf{T}_\Psi}$ - множина переходів, що не містять спільні позиції з іншими, та переходів, вибраних в результаті вирішення конфлікту

$X : T \times \mathbb{R} \rightarrow \{0;1\}$ - предикат, що визначає для кожного переходу приналежність до множини переходів, вибраних в результаті вирішення конфлікту

$$T \in \Psi'(t') \Rightarrow X(T, t') = 1$$

$$T \notin \Psi'(t') \Rightarrow X(T, t') = 0$$

Вхід маркерів в переходи

$$\forall P \in \mathbf{P} \quad M_P(t') = M_P^+(t') - \sum_{T \in P^\bullet \setminus P^\circ} W_{P,T} \cdot X(T, t'), \quad - \text{змінювання стану позиції } P$$

$$\forall T \in \mathbf{T} \mid X(T, t') = 1 \quad E_T(t') = \begin{cases} \{t' + R_T\} \leftarrow \tau_T = \infty \\ E_T^+(t') \cup \{t' + R_T\} \leftarrow \tau_T < \infty \end{cases}$$

- змінювання стану переходу T

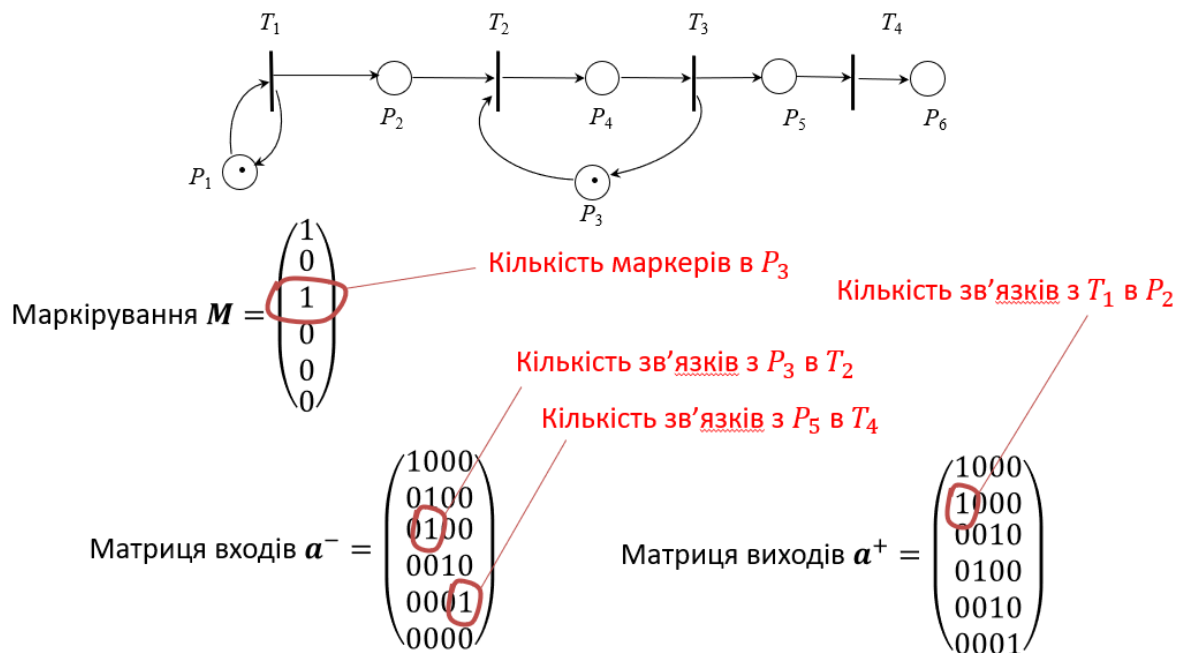
$$D^- : \mathbf{E}(t') \times \mathbf{M}(t') \rightarrow \mathbf{E}(t') \times \mathbf{M}(t') \quad - \text{перетворення стану стохастичної мережі Петрі}$$

$$D^-(\mathbf{S}(t')) \quad - \text{змінювання стану стохастичної мережі Петрі}$$

21. Рівняння станів стохастичної мережі Петрі.

22. Матричні рівняння базової мережі Петрі.

Матричний опис базової мережі Петрі



Матричний опис базової мережі Петрі

Вектор запуску переходу $v = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ Запуск переходу T_3

Умова запуску переходу $\forall i M_i \geq (a^- \cdot v)_i$

Результат запуску переходу $M' = M - a^- \cdot v + a^+ \cdot v$

$$M' = M + (a^+ - a^-) \cdot v$$

Матриця змінювань $a = a^+ - a^-$

Результат запуску послідовності $T_1 - T_2 - T_1$

$$M' = M + a \cdot v^{(1)}$$

$$M'' = M' + a \cdot v^{(2)}$$

$$M''' = M'' + a \cdot v^{(3)}$$

$$M''' = M + a \cdot (v^{(1)} + v^{(2)} + v^{(3)})$$

$$M''' = M + a \cdot v$$

Вектор запуску переходів $v = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 0 \end{pmatrix}$

© І.В.Стеценко НТУУ "КПІ ім. Ігоря Сікорського"

Кількість запусків переходу T_1

Матричне рівняння станів базової мережі Петрі

Результуюче маркірування (після запуску переходів)

Початкове маркірування

$$M' = M + a \cdot v$$

Матриця змінювань

Вектор запуску переходів

Змінювання маркірування

$$\Delta M = a \cdot v$$

23. Матричні рівняння станів стохастичної мережі Петрі.

$$\mathbf{M}(t_n) + \mathbf{a}^+ \cdot \mathbf{v}(t_n) - (\mathbf{M}(t_0) + \mathbf{a}^+ \cdot \mathbf{v}(t_0)) = \mathbf{a}^+ \cdot \boldsymbol{\gamma}(t_n) - \mathbf{a}^- \cdot \boldsymbol{\gamma}(t_n)$$

$$\boldsymbol{\eta}(t_n) = -\mathbf{v}(t_n) + \mathbf{v}(t_0) + \boldsymbol{\gamma}(t_n)$$

где $\mathbf{M}(t) = M_P(t)$ — вектор маркировки позиций в момент времени t ;
 $\mathbf{v}(t) = v_T(t)$ — вектор количества активных каналов переходов; $\mathbf{a}^+ = \|a_{P,T}^+\|$ —
матрица выходов; $\mathbf{a}^- = \|a_{P,T}^-\|$ — матрица входов; $\boldsymbol{\gamma}(t) = \|\gamma_T(t)\|$ — вектор
количества входов в переходы за период времени $[t_0, t]$; $\boldsymbol{\eta}(t) = \|\eta_T(t)\|$ —
вектор количества выходов из переходов за период времени $[t_0, t]$.

$$\boldsymbol{\mu}(t_n) = \boldsymbol{\mu}(t_0) + (\mathbf{a}^+ - \mathbf{a}^-) \cdot \boldsymbol{\gamma}(t_n),$$

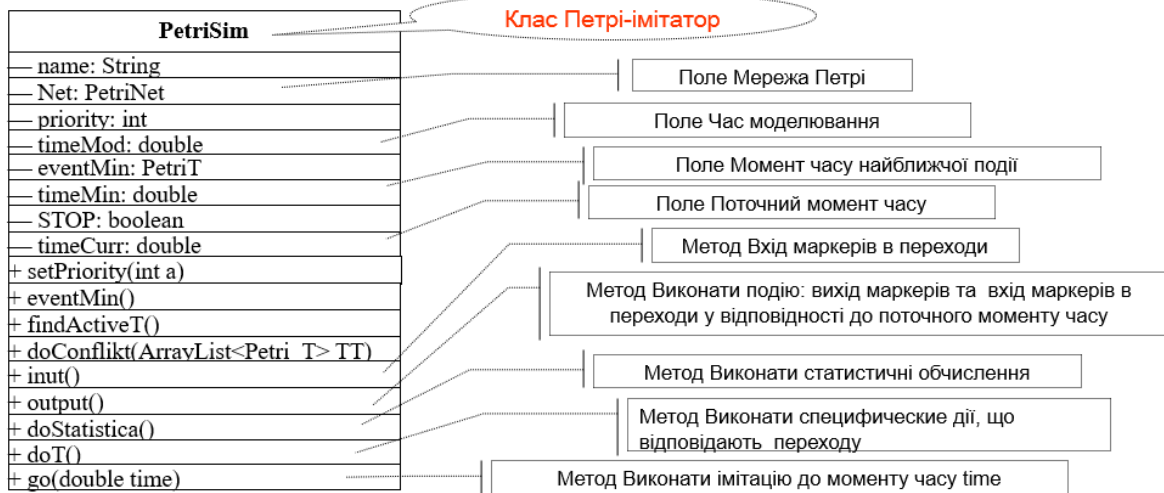
$$\boldsymbol{\eta}(t_n) = -\mathbf{v}(t_n) + \mathbf{v}(t_0) + \boldsymbol{\gamma}(t_n).$$

24. Еквівалентні відношення між базовою, детермінованою та стохастичною мережами Петрі.

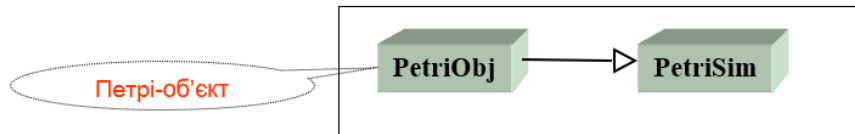
25. Дослідження властивостей мереж Петрі. Інваріанти мережі Петрі.

26. Формалізм Петрі-об'єктної моделі.

Поняття Петрі-об'єкта



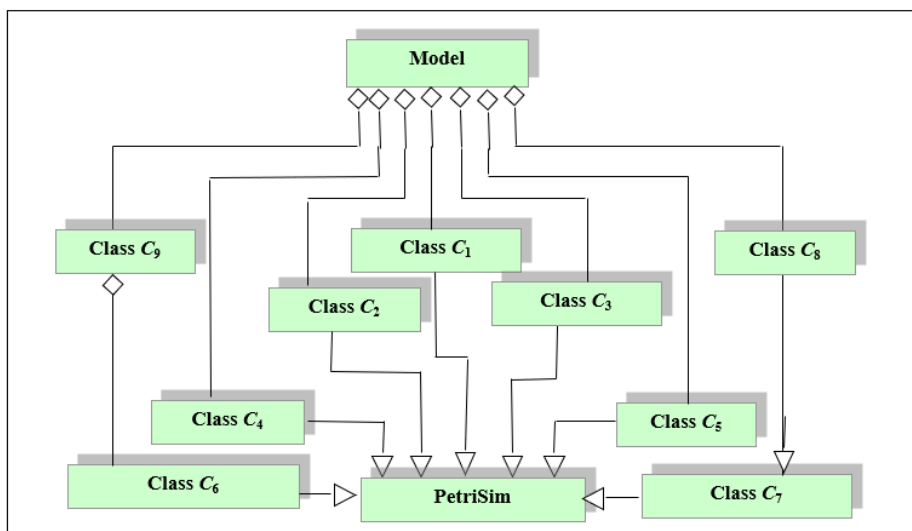
Означення. **Петрі-об'єктом** (PetriObj) називатимемо об'єкт, що є нащадком об'єкта Петрі-імітатор (PetriSim): $\text{PetriObj} \xrightarrow{\text{inherit}} \text{PetriSim}$



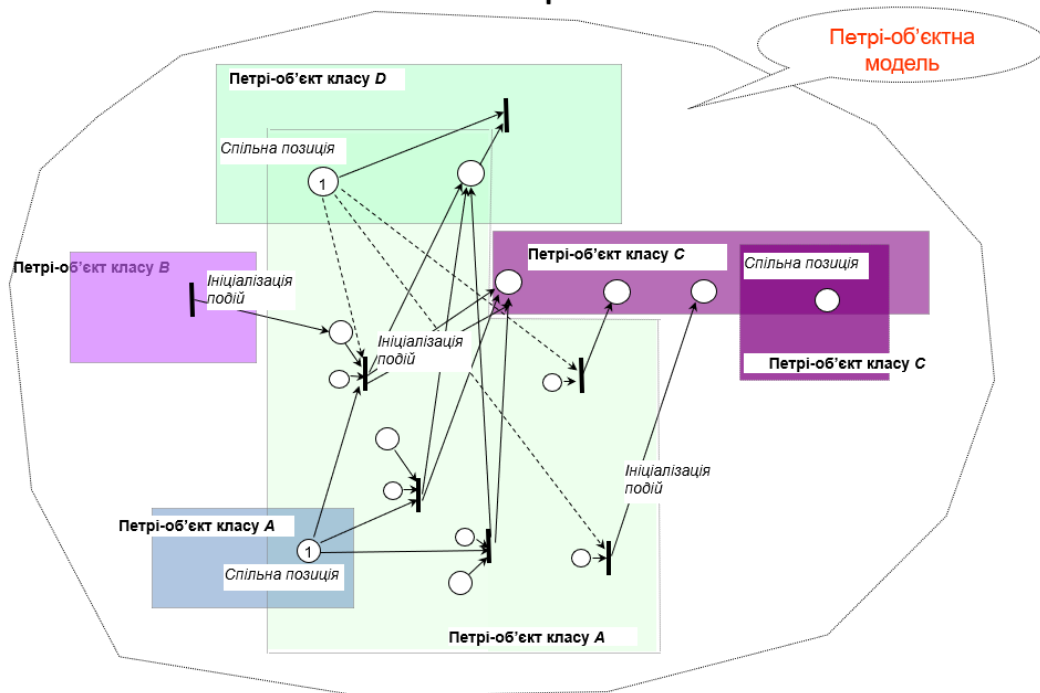
Поняття Петрі-об'єктної моделі

Означення. Петрі-об'єктною моделлю називатимемо модель, що отримана в результаті агрегування Петрі-об'єктів:

$$Model = \bigcup_N O_N, \text{ де } O_N \xrightarrow{\text{inherit}} \text{PetriSim}$$



Зв'язки між Петрі-об'єктами



27. Математичний опис Петрі-об'єктної моделі.

28. Алгоритм імітації Петрі-об'єктної моделі.

Алгоритм імітації Петрі-об'єктної моделі

- Формувати список Петрі-об'єктів;
- Виконати перетворення $(D^-)^m$ (метод input());
- Доки не досягнутий момент завершення моделювання
 - Просунути час в момент найближчої події;
 - визначити список конфліктних об'єктів та вибрати об'єкт із списку конфліктних об'єктів;
 - для вибраного об'єкта виконати перетворення $(D^-)^m \circ D^+$ (методи nextEvent(), doT());
 - для всіх інших об'єктів виконати перетворення $(D^-)^m$ (метод input());
- Вивести результати моделювання.

Анализ обчислювальної складності алгоритму:

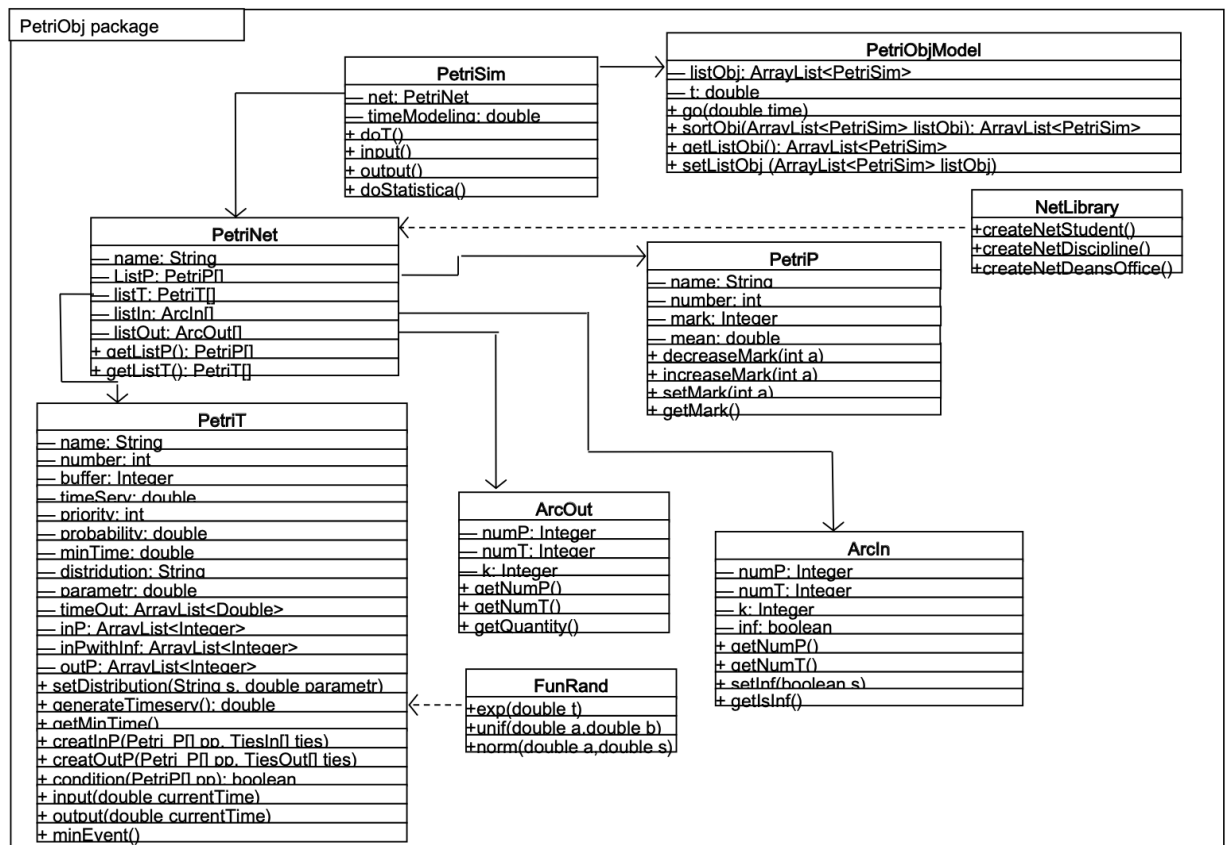
$$O(|T|^2 \cdot V \cdot timeMod \cdot \left(\underset{T \in T}{mean} |T^*| + V + V \cdot |T| \cdot \underset{T \in T}{mean} |T^*| + V^2 \cdot |T| + V \cdot K^2 \right))$$

Середня кількість активних каналів переходів

Середня кількість конфліктних переходів

29. Бібліотека класів Петрі-об'єктного моделювання.

Бібліотека класів Петрі-об'єктного моделювання



30. Програмне забезпечення Петрі-об'єктного моделювання.

31. Моделювання багатопоточних програм стохастичною мережею Петрі.

32. Експериментальні методи дослідження моделей систем.

У багатьох випадках ціллю моделювання є відшукування таких величин або рівнів факторів, при яких відгук моделі досягає оптимальних значень. При імітаційному моделюванні відшукування таких оптимальних значень можливо тільки експериментальними методами.

Для пошуку оптимальних значень використовується метод найшвидкішого підйому. В околі заданої точки будується апроксимація поверхні відгуку моделі за допомогою повного чи дробового факторного експерименту. По збудованій лінійній функції визначається напрямок найшвидкішого підйому до точки оптимуму. Потім робиться невеликий крок у цьому напрямку і процедура повторюється. Поблизу оптимальної точки нахили, визначувані коефіцієнтами рівняння регресії, стають близькими до нуля.

Успіх цього методу сильно залежить від вибору початкової точки і ніколи не можна стверджувати, що знайдений глобальний оптимум, оскільки досліджується тільки локальна область зміни факторів. Через це пошук оптимальних значень факторів потрібно робити при різноманітних початкових точках.

Крок вибирається більшим за розміри області проведення факторного експерименту в початковій точці. Пробні експерименти ставляться в напрямку найшвидкішого підйому доти, поки не буде визначене найліпше значення відгуку моделі в цьому напрямку (рис.7.1). У найліпшій точці знову ставиться факторний експеримент і в напрямку на швидкішого підйому ставляться пробні експерименти. Пошук продовжується доти, доки найліпше значення відгуку моделі не співпадає з точкою, в якій проводився експеримент.

Факторний експеримент

Відгук моделі – вихідна характеристика моделі, що досліджується в експерименті. *Фактор* – вхідна змінна моделі, вплив якої на відгук моделі досліджується в експерименті.

Дослідник задає:

- кількість варійованих факторів k (2, 3, 4 або 5);
- кількість рівнів для кожного фактору q (2 у випадку припущення лінійної залежності відгуку моделі від факторів);
- область змінювання факторів $\Delta x_j, j = 1..k$;
- необхідну точність ε та довірчу ймовірність β вимірювання відгуку моделі y .

На етапі **тактичного планування** експерименту визначають такі умови експерименту, що забезпечать вимір відгуку моделі з заданою точністю та довірчою ймовірністю. У результаті цього етапу планування повинні бути визначені тривалість одного прогону (в одиницях модельного часу) та кількість прогонів p .

На етапі **стратегічного планування** експериментів визначають таку серію експериментів, що забезпечить отримання бажаної інформації про відгук моделі при мінімальних витратах. У результаті цього етапу планування потрібно визначити тип аналізу, який буде використаний, та побудувати матрицю планування відповідно до обраного типу аналізу та заданої кількості факторів.

Тактичне планування факторних експериментів

1. Визначення кількості прогонів

за нерівністю Чебишева

$$p = \frac{\sigma^2}{\varepsilon^2 (1 - \beta)}$$

де σ^2 – дисперсія відгуку моделі, β – довірна ймовірність, ε – точність вимірювання відгуку моделі

за центральною граничною теоремою

$$p = \frac{\sigma^2 \cdot t_{\varphi}^2}{\varepsilon^2}$$

де σ^2 – дисперсія відгуку моделі, ε – точність вимірювання. t_{φ} - аргумент функції Лапласа такий, що ,

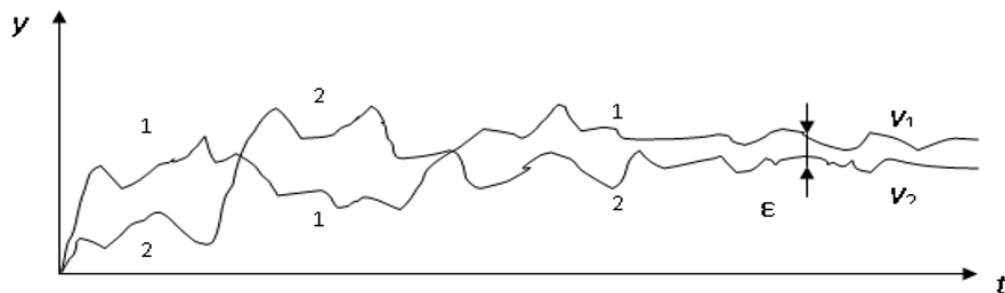
Тактичне планування факторних експериментів

Порівняння оцінки кількості прогонів за нерівністю Чебишева та за центральною граничною теоремою

| ε | $\frac{\sigma^2}{\varepsilon^2(1-0.95)}$ | $\frac{\sigma^2 \cdot 1.96^2}{\varepsilon^2}$ |
|---------------|--|---|
| σ | 20 | 4 |
| $\sigma/2$ | 80 | 15 |
| $\sigma/4$ | 320 | 61 |
| $\sigma/6$ | 720 | 138 |
| $\sigma/8$ | 1280 | 246 |
| $\sigma/10$ | 2000 | 384 |

Тактичне планування факторних експериментів

2. Визначення тривалості одного прогону



Для того, щоб зменшити вплив перехідного періоду на результати моделювання, виконують такі дії:

- вибирають такі початкові умови моделі, що зменшують її перехідний період;
- вилучають значення перехідного періоду при обчисленні середнього значення відгуку моделі;
- використовують такі тривалі прогони, щоб кількість даних, що потрапили в перехідний період, була незначною у порівнянні з кількістю даних сталого стану.

Стратегічне планування факторних експериментів

При дослідженні впливу якісних факторів метою експерименту є якісна оцінка впливу факторів, тобто відповідь на запитання «впливає чи не впливає значення фактора на відгук моделі».

При дослідженні впливу кількісних факторів метою експерименту є кількісна оцінка впливу факторів.

Для якісної оцінки впливу факторів використовують **дисперсійний** аналіз, а для кількісної оцінки впливу факторів - **регресійний** аналіз.

Регресійний аналіз впливу факторів

$$x_i = \frac{X_i - X_{i0}}{\Delta_i} \quad X_{i0} = \frac{X_{i\max} + X_{i\min}}{2} \quad \Delta_i = \frac{X_{i\max} - X_{i\min}}{2}$$

| 2 ³ | x ₀ | x ₁ | x ₂ | x ₃ | x ₁ x ₂ | x ₁ x ₃ | x ₂ x ₃ | x ₁ x ₂ x ₃ | y |
|---|----------------|----------------|----------------|----------------|-------------------------------|-------------------------------|-------------------------------|--|----------------|
| 1 | + | + | + | + | + | + | + | + | y ₁ |
| 2 | + | - | + | + | - | - | + | - | y ₂ |
| 3 | + | + | - | + | - | + | - | - | y ₃ |
| 4 | + | - | - | + | + | - | - | + | y ₄ |
| 5 | + | + | + | - | + | - | - | - | y ₅ |
| 6 | + | - | + | - | - | + | - | + | y ₆ |
| 7 | + | + | - | - | - | - | + | + | y ₇ |
| 8 | + | - | - | - | + | + | + | - | y ₈ |
| $y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_1x_2 + b_5x_1x_3 + b_6x_2x_3 + b_7x_1x_2x_3$ | | | | | | | | | |

Формула обчислення коефіцієнтів b:

$$b_k = \frac{\sum_i y_i x_{ik}}{N}, \quad k = 0, \dots, N-1$$

Статистична обробка результатів факторних експериментів

- 1) Оцінка відтворюваності експерименту
- 2) Оцінка значимості коефіцієнтів регресії
- 3) Оцінка адекватності рівняння регресії
(після відкидання незначимих коефіцієнтів)

33. Еволюційні методи пошуку оптимальних значень параметрів.

34. Програмне забезпечення з імітаційного моделювання.

Найбільш відомі програмні продукти з імітаційного моделювання

- GPSS
- Arena Simulation
- Simio
- SIMUL8
- CPNTools
- PIPE
- AnyLogic
- https://en.wikipedia.org/wiki/List_of_discrete_event_simulation_software

35. Моделювання в Arena Rockwell Software.

система Arena, що створена Кельтоном у 1998 році

Основа технології Arena - мова моделювання SIMAN і система Cinema Animation.

Введення моделі в системі Arena здійснюється простим перетягуванням потрібних модулів із панелі процесів (Basic Process, Advanced Process або Advanced Transfer) у вікно робочого поля.

Параметри модулів, такі як час затримки, вид закону розподілу, кількість пристроїв, кількість

одночасно обслуговуваних об'єктів та інші, задаються у вікні параметрів модулів. Між модулями указуються з'єднання у відповідності до логіки функціонування моделі.

Перед тим, як розпочати процес імітації потрібно зайти в меню Run, підменю Setup і ввести в установки Replication Parameters час моделювання (Replication Length) та одиниці в яких він вимірюється (Time Units), а також одиниці часу, в яких будуть представлені результати моделювання (Basic Time Units). Бажано також ввести в установки Project Parameters ім'я проекту (Project Title) обрати статистичні дані, що збираються (Entities, Queues, Resources, Processes, а також Costing).

Запуск імітації здійснюється вибором команди Go меню Run. Для перевірки моделі зручно скористатись командою Step, що запускає покрокову імітацію модельованого процесу. Управління процесом імітації здійснюють команди Pause - тимчасова зупинка процесу імітації, Fast Forward - швидка імітація, End - повернення до корегування моделі або нового запуску.

В результаті успішної імітації моделі пакетом Arena створюються звіти про результати моделювання. Переглянути їх можна скориставшись панеллю звітів *Reports*. На панелі звітів представлено декілька видів звітів. Звіт «Короткий огляд категорій» і звіти по чотирьох категоріях, такі як Вимоги, Процеси, Черги і Ресурси.

36. Побудова ієрархічних моделей в Arena.

37. Елементи анімації програмного забезпечення Arena.

38. Моделювання в CPNTools.

39. Моделювання в GPSS.

40. Загальні складові сучасного програмного забезпечення з імітаційного моделювання систем.

Основні компоненти ПЗ імітаційного моделювання дискретно-подійних систем

- Комплекс елементів для складання моделей.
- Графічний редактор для складання, редагування та збереження моделі з набору елементів.
- Модуль для збору статистичних даних (стандартний або з можливістю налаштування).
- Модуль для управління збором статистики (час моделювання, час розгону моделі, налаштування набору даних, по яких збирається статистика).
- Модуль для проведення експериментальних досліджень з моделлю (дослідження відгуку в часі, факторний експеримент, експеримент з метою оптимізації).
- Анімація процесу функціонування моделі (2D, 3D).
- Тестові моделі для демонстрації імітаційного моделювання.
- Модуль для аналізу властивостей моделі.

Сучасні тенденції розвитку ПЗ імітаційного моделювання

- Підтримка паралельної реалізації алгоритму імітації (одно чи багатопроцесорної)
- Підтримка віддаленої роботи з моделлю системи (хмарні технології)
- Предметно-орієнтовані середовища моделювання ([VisSim](#), Network Simulator)

41. Паралельні обчислення в імітаційному моделюванні.
42. Паралельний алгоритм імітації Петрі-об'єктної моделі.