

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання самостійної роботи
з дисципліни
“Моделювання систем”
для студентів
спеціальності 6.05010102
“Інженерія програмного забезпечення”
усіх форм навчання

2017

Методичні вказівки до виконання самостійної роботи з дисципліни “Моделювання систем” для студентів спеціальності 6.05010102 “Інженерія програмного забезпечення” усіх форм навчання /Укл.: С.М. Сердюк, Ж.К. Камінська. – Запоріжжя: ЗНТУ, 2017. – 41с.

Укладачі: С. М. Сердюк, доцент, к.т.н.,
Ж.К. Камінська, асистент

Рецензент: С.К. Корнієнко, доцент, к.т.н.

Відповідальний
за випуск: С.М. Сердюк, доцент, к.т.н.,

Затверджено
на засіданні кафедри
“Програмних засобів”

Протокол № _____
від _____ 2017

ЗМІСТ

1 Мета впровадження самостійної роботи студентів під керівництвом викладача в процесі навчання.....	3
2 Загальні відомості про дисципліну.....	5
3 Теоретичні питання з дисципліни.....	7
4 Перелік лабораторних робіт.....	9
5 Технологія використання системи імітаційного моделювання SERVICEMODEL	10
6 Приклади індивідуальних завдань.....	40
ЛІТЕРАТУРА.....	41

1 МЕТА ВПРОВАДЖЕННЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ ПІД КЕРІВНИЦТВОМ ВИКЛАДАЧА В ПРОЦЕСІ НАВЧАННЯ

Якість професійної підготовки фахівця визначається вмінням самостійно реалізовувати виробничі та наукові задачі, готовністю до самоосвіти. Такі властивості фахівець повинен набути під час навчання у вищому навчальному закладі. Особливо це стосується випускників кафедри «програмне забезпечення», які працюють у різних галузях науки і виробництва та повинні орієнтуватися у ринку програмних продуктів, який швидко змінюється.

Тому метою навчального процесу є не тільки набуття знань, а і розвиток пізнавальної самостійності студента і його творчих здібностей.

Досягнення вказаної мети навчального процесу можливо, якщо всі форми навчання організовано з врахуванням цієї мети. Самостійна робота включає до себе всі основні види навчальних занять: проблемні лекції, практичні заняття, лабораторні роботи, семінари, поза аудиторну роботу.

Метою самостійної роботи з навчальної дисципліни «Моделювання систем» є поглиблення та закріплення знань з методів дослідження комп'ютерних інформаційних систем, а також практичне застосування знань, освоєних при виконанні завдань на самопідготовку, вивченні забезпечувальних та суміжних навчальних дисциплін, а також в ході лекційних і лабораторних робіт.

В методичних вказівках викладаються основні поняття та технологія імітаційного моделювання інформаційних систем за допомогою системи імітаційного моделювання (CIM) ServiceModel.

2 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ДИСЦИПЛІНУ

2.1 Мета викладання дисципліни

Метою дисципліни є формування у студентів знань з основ складання моделей систем різних класів, дослідження цих моделей і обробки результатів досліджень, використовуючи інструментальні засоби імітаційного моделювання.

2.2 Задачі вивчення дисципліни

Внаслідок вивчення дисципліни студенти повинні:

- бути підготовленим до участі у всіх фазах дослідження, проектування, розробки, виготовлення та експлуатації систем;
- використовувати основні закони природничих дисциплін у професійній діяльності, застосовувати методи математичного аналізу і моделювання, теоретичного та експериментального дослідження систем;
- володіти основними принципами і методами побудови (формалізації) і дослідження математичних моделей систем, їх формами подання та перетворення;
- володіти основними програмними засобами, що використовуються при моделюванні;
- володіти основними прийомами обробки та представлення експериментальних даних.

На основі вивчення дисципліни "Моделювання систем" студенти повинні **знати**:

- методологічні основи моделювання;
- принципи математичного та імітаційного моделювання систем;
- методи та етапи дослідження моделей систем;
- основи планування експериментів з такими моделями;
- статистичні методи обробки результатів експериментів.

вміти:

- використовувати методи математичного моделювання при розробці систем;

- вибирати раціональний підхід до моделювання системи, оцінювати точність і об'єктивність моделі, моделювати вплив на систему, саму систему та її відгук на вплив;
- здійснювати синтез систем на підставі аналізу результатів моделювання;
- на практиці застосовувати інструментальні засоби імітаційного моделювання.

2.3 Загальні вказівки

Дисципліна вивчається один семестр. Студенти повинні систематично вивчати та закріплювати теоретичний матеріал. Для цього студент самостійно працює над програмою дисципліни за вказаною літературою і матеріалом лекцій, після чого обов'язково виконує лабораторні та індивідуальні завдання за темою. За розкладом викладач організує консультації по вивченню теоретичного матеріалу і виконанню завдань за темами.

Знання, які одержано під час самостійної роботи, необхідні для глибокого розуміння і розширення матеріалу, що вивчався на лекціях. Рівень знань за темою та оцінювання результатів виконання завдань перевіряється під час захисту лабораторних робіт та іспиту.

3 ТЕОРЕТИЧНІ ПИТАННЯ З ДИСЦИПЛІНИ

3.1 Основи теорії моделювання систем

Моделювання як метод наукового пізнання. Використання моделювання при дослідженні і проектуванні складних систем. Поняття системи. Опис елементів системи, зв'язків та їх сукупностей. Опис з'єднання елементів і формування матриць зв'язків. Опис системи як сукупності зв'язаних елементів. Принципи системного підходу.

Література [1, стор.4-16; 6, стор. 90-119; 7, стор.14-35; 8, стор.20-27; 10, стор.37-41, 65-67; 11, стор.5-9, 20-44].

3.2 Структуризація систем

Стратифікований підхід до моделювання систем Месаровича. Його недоліки. Узагальнений підхід до структуризації ерготехнічних систем Губінського. Множина структур необхідних для проведення системного аналізу та формалізації систем: компонентних, морфологічних та еволюційних структур.

Література [9, стор.37-41, 65-67].

3.3 Класифікація видів моделювання систем

Необхідність класифікацій моделей. Поняття детермінованих та стохастичних моделей. Основні положення теорії подібності. Статичні, динамічні, дискретні, дискретно-безперервні та безперервні види моделювання. Основні характеристики видів моделювання. Імітаційні моделі систем.

Література [1, стор.21-28].

3.4 Дискретні системи моделювання

Узагальнена структура та функціонування систем моделювання. Організація квазіпаралелізму в моделях засобами розгляду активності. Особливості активностних, транзактних, агрегатних, процесно-орієнтованих систем моделювання. Моделювання систем масового обслуговування.

Література [5, стор.14-36; 12, стор. 105-118].

3.5 Технологія проектування імітаційних моделей

Етапи створення імітаційних моделей: змістовне описання об'єкту моделювання; розробка концептуальної моделі; перехід до формального опису; перетворення формальної мови в описання імітаційної моделі; програмування та відлагодження імітаційної моделі. Проведення імітаційного експерименту.

Література [5, стор.37-91; 9, стор.35-40, 57-64; 11, стор. 144-197].

3.6 Призначення та види мов моделювання

Необхідність та переваги мов моделювання. Дискретні, безперервні та дискретно-безперервні мови імітаційного моделювання. Засоби підвищення рівня технології побудови моделей. Особливості аналізу результатів моделювання. Перспективні архітектури середовищ для підтримки імітаційного моделювання.

Література [5, стор.99-120; 6, стор.206-225; 11, стор. 144-197]

4 ПЕРЕЛІК ЛАБОРАТОРНИХ РОБІТ

4.1 Лабораторна робота №1. Методи створення моделей з використанням системи імітаційного моделювання SIMC.

Мета роботи: ознайомлення з принципами побудови імітаційних моделей систем масового обслуговування (СМО) з використанням системи імітаційного моделювання SIMC.

4.2 Лабораторна робота № 2. Моделювання систем масового обслуговування з одним обслуговуючим приладом та чергою

Мета - вивчення методів моделювання різних дисциплін обслуговування в СМО з одним приладом та чергою, аналіз вихідних даних моделювання з метою вибору оптимального варіанту реалізації системи яка моделюється.

4.3 Лабораторна робота № 3. Моделювання системи масового обслуговування зі зворотнім зв'язком. Моделювання багатоканальної системи масового обслуговування.

Мета роботи: вивчення методів моделювання СМО із зворотнім зв'язком та багатоканальних СМО на основі використання систем імітаційного моделювання SIMC.

4.4 Лабораторна робота № 4. Використання розподілу імовірності в системі імітаційного моделювання SIMC. Генератори випадкових чисел.

Мета роботи: вивчення методів описання рівномірного та нерівномірного розподілу безперервних та дискретних випадкових величин в SIMC та способів їх практичного використання при моделюванні СМО.

4.5 Лабораторна робота № 5. Моделювання довільних дисциплін обслуговування з використанням ланцюгів користувача. Вивчення принципів побудови гістограм.

Мета роботи: освоєння процедур графічного виводу інформації про розподіл випадкових величин та синхронізації подій в моделі (сигнали) в системі імітаційного моделювання SIMC. Вивчити принципи моделювання довільних дисциплін обслуговування на основі списків користувач.

4 ТЕХНОЛОГІЯ ВИКОРИСТАННЯ СИСТЕМИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ SERVICEMODEL

4.1 Короткі теоретичні відомості

ServiceModel - це **Microsoft Windows** базовий імітаційний програмний пакет, розроблений для моделювання систем масового обслуговування, за допомогою котрого можливо ефективно вирішувати реальні задачі, такі як:

- **Process Reengineering** - реорганізація виробництва;
- **Work Flow Analysis** - аналіз робочих потоків;
- **Facility Planning** - планування ресурсів;
- **Office Automation** - автоматизація офіса;
- **Capacity Analysis** - аналіз продуктивності;
- **Expansion Planning** - планування розширення.

ServiceModel являє собою систему для імітації процесів на робочих місцях у вигляді обслуговування черг по обробці різних об'єктів (документів, виробів, клієнтів і т.п.).

Процеси на робочих місцях підкоряються певній логіці, що може описувати як взаємодію вхідних об'єктів, так і умову передачі цих об'єктів на наступні робочі місця. СІМ підтримує зв'язки між робочими місцями, які описують їх відношення, а також враховуються ресурси, необхідні для виконання робочих процесів(люди і/або устаткування, програмне забезпечення і т.і.).

Рівень і подробиці опису моделі визначаються цілями моделювання. Під робочими місцями можна розуміти: цілком компанію з зовнішніми організаціями (якщо важливо моделювати роботу відносно партнерів і замовників); підрозділи компанії (якщо необхідно знати взаємодію між підрозділами); усі робочі місця у всіх підрозділах (включаючи місця відпочинку, прийняття їжі та ін.); або усе разом.

4.2 Основні поняття системи моделювання

Основними поняттями моделювання СІМ **ServiceModel** є:

- робочі місця (**locations**)
- об'єкти обробки(**entities**)

- ресурси(**resources**)
- процеси на робочих місцях(**processing**):
- опис процесу(**operation**)
- зв'язки між робочими місцями(**routing**)
- маршрути переміщення (**path networks**)
- параметри вхідних потоків(**arrivals**)

Модель будь-якого виробничого процесу може бути подана у вигляді сукупності робочих місць, зв'язаних із ними ресурсів, об'єктів які оброблюються на цих місцях і логічних відношень між робочими місцями. Логічні відношення визначають алгоритми послідовної обробки об'єктів, що повинні описувати реально існуючі або плановані виробничі процеси.

Параметри вхідних потоків (*arrivals*)

Параметри вхідних потоків (рис.4.1) визначають вихідні потоки для моделі які мають наступні характеристики:

- кількість екземплярів об'єктів обробки в порції (n);
- кількість порцій (q);
- час надходження першої порції (t_0);
- час між порціями (τ).

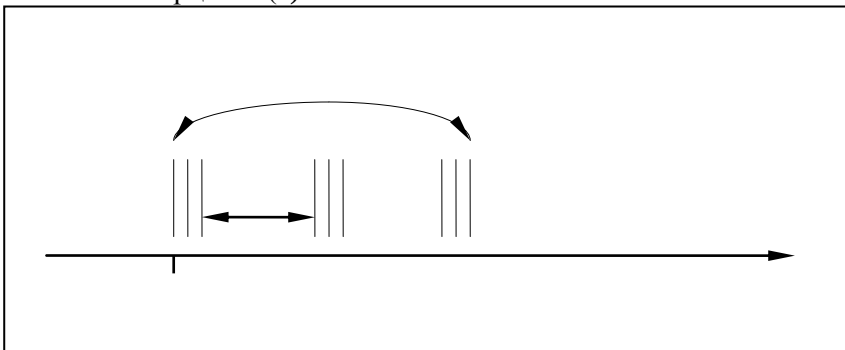


Рисунок 4.1 – Параметри вхідних потоків

Кожний із параметрів може визначатись формулою з використанням наступних дистрибутивних законів розподілу :

Uniform - Постійний;

Triangular - Трикутний;

Normal - Нормальний;

Exponential - Експоненціальний;
Weibull - Вейбулла;
Erlang - Ерланга;
Beta - Бетта;
Lognormal - Логнормальний;
Inverse Gaussian - Інверсний Гауссовському;
Pearson5 - Пірсон5;
Pearson6 - Пірсон6;
Binomial - Біноміальний;
Geometric - Геометричний;
Poisson - Пуассоновський.

Крім того, опис вхідного потоку може бути заданий в табличному вигляді, який описує реальну статистику.

Таблиці, що описують вхідні потоки можуть бути підготовлені за допомогою зовнішніх програм (наприклад, у *MS Excel*).

Робочі місця (locations)

Робоче місце (рис.4.2) визначається функцією (операцією) конкретного бізнес-процесу і має:

- назву;
- зображення;
- ємність, що визначає кількість об'єктів, які можуть одночасно знаходитися в даному робочому місці;
- кількість одиниць даного робочого місця;
- правила обробки об'єктів, що містять:
- правило вибору об'єктів для поточної обробки з потоку;
- правило формування вихідного потоку;
- правило розподілу обробки вхідних об'єктів по робочим місцям, якщо місць даного типу більше одного;
- текстовий опис робочого місця.



Рисунок 4.2 – Приклади робочих місць

Об'єкти обробки (рис.4.3) є елементами потоків, що обробляються на робочих місцях. Кожний об'єкт має:

- назву;
- зображення (одне або багато);
- швидкість руху об'єктів у м/мин або фут/мин між робочими місцями;
- текстовий опис особливостей.

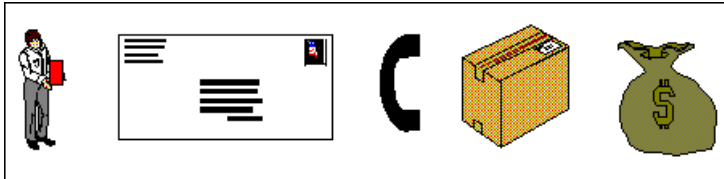


Рисунок 4.3 – Приклади об'єктів обробки

Ресурси (resources)

Ресурси - це усе, що забезпечує виконання операцій над об'єктами (службовці, обчислювальна техніка, верстати, оргтехніка, транспорт і т.д.).

Ресурси (рис.4.4) можуть мати наступні атрибути:

- назву;
- зображення;
- кількість одиниць;
- опис характеристики пересування і локалізації ресурсів (маршрути, місця зосередження, швидкості пересування, затримки й ін.);
- опис логіки поведінки ресурсів;
- текстовий опис особливостей використання ресурсів.

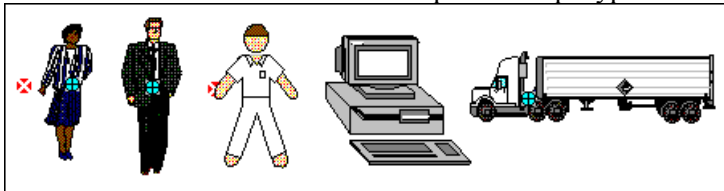


Рисунок 4.4 – Приклад ресурсів

Процеси на робочих місцях (processing)

Процеси на робочих місцях описують алгоритми обробки об'єктів і є вузловими моментами поведінки моделі в цілому. Є наступні можливості опису процесів:

- типи об'єктів, підлягаючих обробці на даному робочому місці;
- назва робочого місця, де відбувається даний процес;
- опис алгоритму обробки з використанням елементів програмування (*IF-THEN*, *DO WHILE*, *GO_TO* та інших вбудованих функцій або стверджень), із вказівкою часу продовження процесу і переліку робочих місць для наступної обробки об'єктів;
- опис кожного наступного шляху передачі об'єктів з вказівкою правил та способів передачі об'єктів по кожному з них.

Маршрути переміщення (path networks)

Маршрути переміщення описують переміщення ресурсів і об'єктів обробки при виконанні зв'язаних з ними операцій. Переміщення людських ресурсів може бути зв'язано з підходом для виконання операцій до робочого місця від місць відпочинку, прийняття їжі, перекурів, і т.п., а також відходів від робочого місця до місць відпочинку і/або іншої локалізації.

Кожний такий маршрут характеризується часом, що є, у загальному випадку, не продуктивним (загубленим).

Маршрути задаються графічно в метриці розташування робочих місць.

Додаткові можливості

Опис регламенту - розклад використання ресурсів або розклад готовності робочих місць.

Розклад описується по годинам днів тижня і визначає коли може бути використан той або інший ресурс або коли активне робоче місце.

Таких розкладів може бути декілька, кожен з них прив'язується до конкретного ресурсу або робочого місця.

Розклад задається в табличному вигляді. Його приклад наведений на рисунку 4.5.

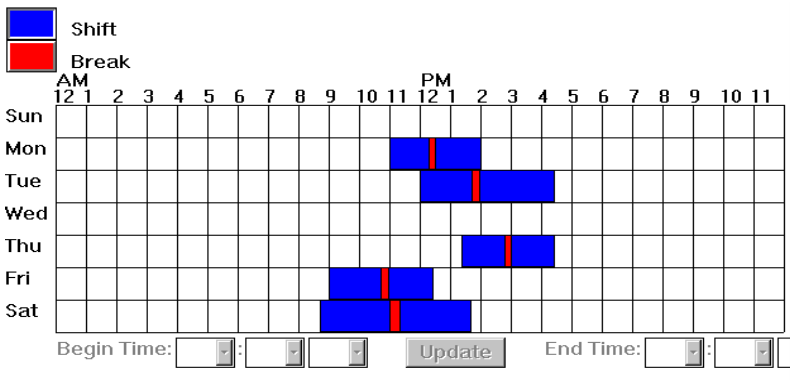


Рисунок 4.5 – Приклад розкладу

Опис сценарію - можливість моделювати кілька сценаріїв для моделі процесу, що відрізняються набором параметрів (кількістю ресурсів, коефіцієнтом розподілу потоку). Приклад завдання сценарія наведений на рисунку 4.6.

Scenario Parameters

Scenario Name:

Parameter	Current Setting
Mean Arrival Quantity	75
Auditor Processing Time	N(20,5)
Loan Processing Time	N(15,3)
Service Processing Time	N(12,3)
Cross Trained Staff?	2

Рисунок 4.6 – Приклад завдання сценарія

Сценарії допомагають досліджувати кілька варіантів поведінки моделі процесу з метою наступного вибору найбільш оптимального.

Вікно ServiceModel та його елементи

Після запуску **ServiceModel** на екрані відображається головне вікно програми. Структура екрана при роботі з **ServiceModel** багато в чому аналогічна структурі екрана при роботі з іншими Windows-додатками.

Рядок меню

Під рядком заголовка знаходиться рядок меню (рис.4.7), у якому містяться усі команди CIM **ServiceModel**.



Рисунок 4.7 – Меню команд CIM ServiceModel

File

- New - створення нової моделі;
- Open - відкриття існуючої моделі;
- Merge - об'єднання моделей;
- Save - вивантаження моделі;
- Save As... - зберегти як... ;
- View Text - перегляд параметрів моделі;
- Print Text - друкує параметри моделі;
- Print Setup - налаштування принтера;
- Create Model Package... - створення модельного пакета;
- Install Model Package... - установка модельного пакета;
- Exit - вихід.

Build

- Locations - робочі місця;
- Entities - об'єкти обробки;
- Path Networks - маршрути переміщення;
- Resources - ресурси;
- Processing - процеси на робочих місцях;

Arrivals - параметри вхідних потоків;
 Shifts - регламенти;
 Define - опис регламенту;
 Assign - призначення регламенту;
 More elements - додаткові елементи;
 Attributes - атрибути;
 Variables(global) - глобальні змінні;
 Macros - макроси;
 Subroutines - підпрограми;
 Arrival cycles - цикли вхідних потоків;
 Table function - табличні розподіли;
 User definitions - визначення користувача;
 External files - зовнішні файли;
 General Information - основна інформація про модель;
 Background Graphics - фонові графіки;
 Front of Grid - над умовними позначеннями;
 Behind Grid - під умовними позначеннями.

Simulation

Run - запуск моделі;
 Save & Run - зберігання і запуск моделі;
 Options - опції роботи моделі;
 Model parameters - параметри моделі;
 Scenarios - сценарії.



Контрольні питання

1. Що становити собою модель процесу в CIM **ServiceModel**?
2. Назвіть параметри вхідних потоків.
3. Сформулюйте перелік атрибутів поняття "робоче місце".
4. Сформулюйте перелік атрибутів поняття "об'єкти обробки".
5. Що таке ресурси? Які вони мають атрибути?
6. Назвіть можливості опису процесів.
7. Що таке розклад? Як він задається?

4.3 Технологія моделювання в ServiceModel

При спробі створення нової моделі на екран буде виведен запит основних даних моделі (General Information). У цьому вікні задається назва моделі (не ім'я файла), одиниці виміру часу та відстані, графіка моделі (колір фону, колір і крок масштабної сітки, файл з графічними даними), початкова логіка і логіка завершення, а також примітки. Встановивши необхідні значення (або залишивши значення по умовчанню), можна приступати безпосередньо до моделювання.

Порядок моделювання довільний. Єдиною умовою є повнота і цілісність моделі, тобто для кожного об'єкта, що увійшов у модель, повинні бути визначені його дії на шляху його проходження до виходу.

Кожна модель повинна містити робочі місця (locations) і об'єкти обробки (entities). У кожній моделі повинні визначатися параметри вхідних потоків (arrivals) для об'єктів обробки (якщо не визначені вхідні потоки, то жодний об'єкт у модель не потрапить) і процеси на робочих місцях (processing), тобто як і що будуть робити об'єкти в моделі. Всі інші елементи - ресурси, сценарії, атрибути і т.д. - по суті є необов'язковими.

4.3.1 Приклад створення простої моделі за допомогою ServiceModel

Створимо модель найпростішого процесу: покупець підходить до кіоска, купує газету і відходить.

File|New

У вікні General Information дайте назву моделі (наприклад "Кіоск"). Інші значення можна залишити по умовчанню. Натисніть <OK>. Ви увійшли в середовище редагування моделі.

Build|Locations

Визначаємо робочі місця (у нашому випадку - один кіоск). У вікні Graphics (рис.4.8) вибираємо графічне представлення об'єкта і тиснемо лівою кнопкою миші у вікні Layout - на тому місці з'являється графічне зображення нашого кіоска.

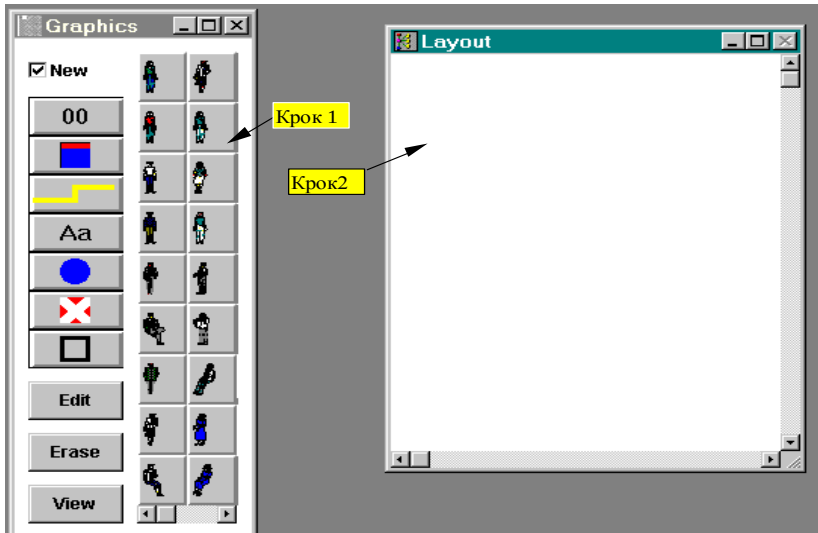


Рисунок 4.8 – Створення робочого місця

У той же час у вікні Locations з'являється рядок з описом об'єкта, у якому присутні наступні параметри (рис.4.9):

- Графічне представлення (Icon)
- Ім'я (Name)
- Ємність (Cap)
- Кількість одиниць робочого місця (Units)
- Визначення режиму виходу з роботи (DTs)
- Збір статистики по робочому місцю (Stats)
- Правила вибору вхідних об'єктів (Rules)
- Примітки (Notes)

Для простоти залишимо всі значення по умовчанням.

Build|Entities

Визначаємо об'єкти обробки (у нашому випадку - покупець). У вікні Entity Graphics вибираємо зображення. При цьому в таблиці Entities автоматично з'являється рядок з описом типу даних об'єктів обробки. Зауважте, що зображення не варто поміщати в поле моделі, як це потрібно у випадку з робочими місцями, тому що об'єкти не

будуть нерухомо стояти в полі, а будуть переміщатися. Рядок опису типу об'єктів містить наступні поля (рис.4.10):

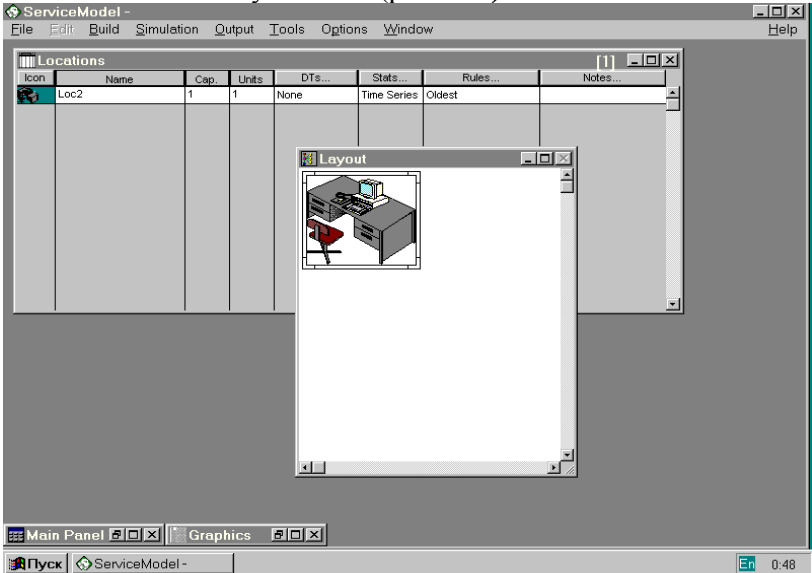


Рисунок 4.9 – Робоче місце та його опис

- Графічний вигляд (Icon)
- Ім'я (Name)
- Швидкість (Speed, у скобках наведені одиниці виміру швидкості)
- Збір статистики по типу об'єктів (Stats)
- Примітки (Notes)

Необхідно відзначити, що в таблиці задаються типи об'єктів, а не самі об'єкти. Іншими словами, у моделі може знаходитися одночасно декілька об'єктів того самого типу. У нашому випадку, як і раніше, залишимо всі параметри об'єкта (людини) незмінними.

Build|Processing

Визначаємо дії об'єктів (над об'єктами) у системі. У нашому випадку ми можемо представити весь процес однією дією - покупкою газети у кіоску (затримкою). Всі пересування по моделі прив'язуються до безпосередніх дій на місцях обробки. У першому рядку вікна Process у полі Entity виберемо ім'я нашого об'єкта

(точніше, ім'я типу). У полі Location вибираємо ім'я робочого місця (кіоска).

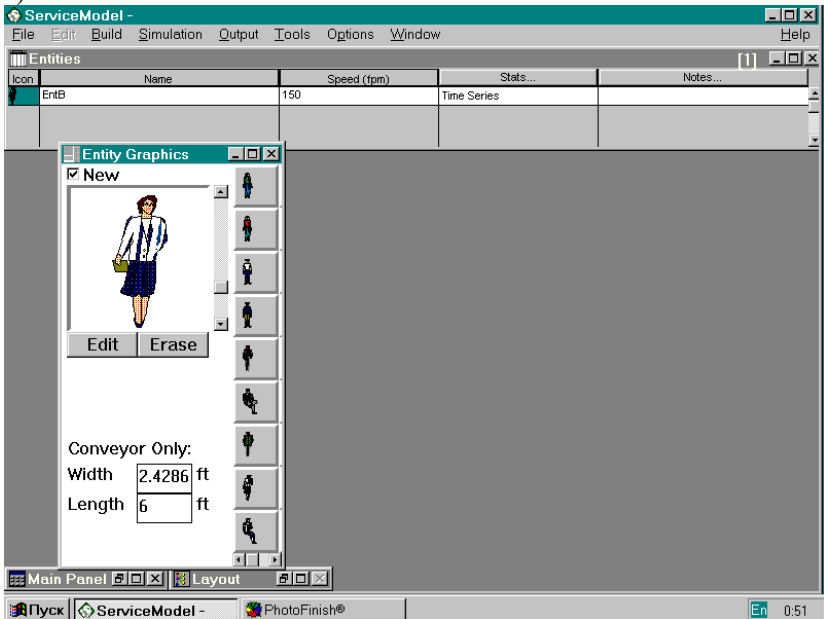


Рисунок 4.10 – Визначення об'єктів обробки

У полі Operation описуємо, що буде відбуватися з покупцем у кіоска. Для простоти змусимо його там затриматися на 5 одиниць часу - Wait 5. Перейдемо у вікно Routing. У ньому визначаємо, куди направиться об'єкт після обробки в кіоску. Можливих варіантів подальших переміщень може бути декілька (у залежності від умови, завантаженості пункту призначення і т.і.). Таким чином, кожному рядку у вікні Process буде відповідати своє вікно Routing, що містить декілька рядків. Опишемо колонки вікна Routing:

- Варіант переміщення (Blk)
- Тип вихідного об'єкта (Output)
- Пункт призначення (Destination)
- Правило маршрутизації (Rule)
- Логіка переміщення (Move Logic)

У нашій моделі покупець який купив газету відходить від кіоска, отже, виходить із моделі. Тому в поле Destination вікна

Routing ставимо EXIT. Усі інші поля ролі не грають. Таким чином, доля кожного покупця, що підходить до кіоска, визначена, тому перейдемо до визначення вхідних потоків.

Build|Arrivals

Визначаємо вхідні потоки об'єктів обробки. У вікні Arrivals подана таблиця з наступними полями:

- Тип об'єктів (Entity)
- Місце входу в модель (Location)
- Кількість екземплярів об'єктів, що надходять у модель за одну порцію (Qty each)
- Момент надходження першої порції (First Time)
- Кількість порцій усього (Occurrences)
- Частота надходження порцій (Frequency)
- Логіка надходження в модель (Logic)
- Вмикання/відключення потоку (Disable)

Задамо достатній набір параметрів для вхідного потоку об'єктів (покупців) до пункту входу в модель (кіоск). Вписуємо ім'я типу об'єктів у поле Entity, ім'я кіоска в поле Location, Qty each - 1, First time - 0, Occurrences - 10, Frequency - 8. Ці параметри однозначно задають вхідний потік людей. Таким чином, починаючи з моменту часу 0, до кіоска (у модель) будуть підходити покупці (об'єкти) через кожні 8 одиниць часу. Усього до кіоска підійде 10 чоловік.

Вигрузіть модель (F12) і запустіть на виконання (F10). Виконання завершиться, коли буде вичерпаний вхідний потік, тобто коли усі 10 покупців придбають по газеті і підуть далі по своїх справах. По закінченню роботи програма запитає, чи варто збирати статистику, і чи хочете ви подивитися результати роботи. При підтвердженні буде згенерован звіт та запущен модуль перегляду звітів (ServiceModel Output).

Побудована модель може послужити основою для опису більш складної системи. Однак із збільшенням масштабів системи процес моделювання і програмування стає усе більш громіздким. Тому вітається покроковий підхід - поступово добавляйте нові елементи і ретельно тестуйте нову модель на кожному кроці.

4.3.2 Види звітів

Перегляд вихідних результатів моделі можливий у двох випадках:

- після закінчення роботи моделі;
- за допомогою модуля “Перегляд статистики” (View statistics) у меню Output.

Вивід результатів моделі зводиться в генеральний текстовий звіт (General report), що складається з усіх даних, які безпосередньо використовувались при моделюванні. Однак, інформація з вихідними результатами у цій (текстовій) формі не є зручною для перегляду. ServiceModel представляє декілька графічних звітів для зручності користувача.

Наступною цікавою можливістю ServiceModel є наявність опції створення вихідних файлів, які можливо відкривати під час порівняння різних моделей. Для створення таких файлів необхідно вибрати модуль “Open Multiple” у меню File і додати поточний вихідний файл.

4.3.3 Методика розробки моделей за допомогою пакета ServiceModel.

Проведемо аналіз стандартної моделі, що поставляється з інсталяційним пакетом ServiceModel. Розглянемо першу фазу розробки моделі деякого офіса (... \models\training\phase1. mod). Відкрийте модель (**File|Open**) - на екрані з'явиться план офіса (рис.4.11).

Стрілки показують маршрути руху об'єктів по моделі і відповідають записам у вікні Routing при визначенні процесів (Build|Processing).

Відкриємо таблицю властивостей робочих місць (Locations) (рис.4.12).

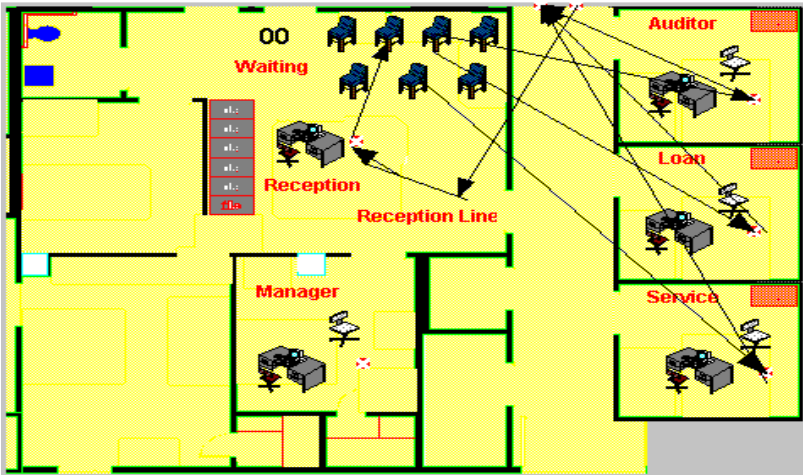


Рисунок 4.11 – План офіса (вікно Layout)

Locations							
Icon	Name	Cap.	Units	DTS...	Stats...	Rules...	Notes...
	Enter	Inf	1	None	Time Series	Oldest	
	Depart	Inf	1	None	Time Series	Oldest	
	Recep_Line	5	1	None	Time Series	Oldest	
	Reception	1	1	None	Time Series	Oldest	
	Waiting	7	1	None	Time Series	Oldest	
	Auditor	1	1	None	Time Series	Oldest	
	Loan	1	1	None	Time Series	Oldest	
	Service	1	1	None	Time Series	Oldest	
	Manager	1	1	None	Time Series	Oldest	

Рисунок 4.12 – Таблиця робочих місць

У моделі присутньо 9 робочих місць:

- Enter - вхід в офіс;
- Depart - вихід з офіса;
- Recep_Line - черга;
- Reception- секретар у приймальні;
- Waiting - зал чекання;
- Auditor - ревізор;
- Loan - фахівець з позик;

- Service - бюро послуг;
- Manager - менеджер.

Кожне з робочих місць має свої параметри. Зарезервоване слово `inf` позначає нескінченність, тобто ємність входу і виходу з офіса необмежена. Виберіть один з рядків таблиці - у вікні Layout буде виділене відповідне йому робоче місце. В одне робоче місце можуть входити декілька графічних образів. Так, робоче місце Auditor містить у собі стіл, стілець, напис і значок. Робоче місце Waiting перебуває з семи стільців, лічильника і напису. Додавання зображень до того ж самого робочого місця робиться так само, як і оголошення нового (див. п. 4.3.1), але при цьому у вікні Graphics необхідно очистити поле New.

Перейдемо до об'єктів обробки (Build|Entities). У даній моделі маємо усього лише один тип - Client. Швидкість переміщення наших клієнтів по моделі - 150 футів у хвилину.

Далі подивимось на вхідні потоки (Build|Arrivals). У нашому випадку ми маємо справу лише з одним потоком: клієнти надходять у модель по одному, починаючи з моменту часу 0 у пункт Enter. Усього в систему надійде 50 клієнтів. Час надходження підкоряється експоненціальному закону із середнім значенням 10 хвилин. Далі з'ясуємо, що відбувається з клієнтами в самій моделі безпосередньо.

Відкриваємо таблицю процесів (Build|Processing). У цей же час на плані офіса вимальовуються стрілки, що показують переміщення об'єкта по моделі. Стрілка поточного (виділеного) процесу виділена окремим кольором. Простежимо рух клієнтів по моделі:

- з робочого місця Enter клієнт рухається до черги Reser_Line. Час переміщення - півхвилини.

- з пункту Reser_Line клієнт направляється до секретаря (Reception), причому не затрачає на це ніякого часу (MOVE FOR 0). Таким чином, Reser_Line є буфером чекання (чергою) перед столом секретаря.

- підійшовши до секретаря, клієнт затримується на визначений час, нормально розподілений в інтервалі між 1,7 і 2,3, а потім переміщується в зал чекання (Waiting). Процес переходу в зал займає у людини 0,1 хвилини.

- процес для залу чекання декілька відрізняється від попередніх, тому що в ньому передбачається розгалуження подальшого пересування клієнта по моделі. Вікно Routing for Client @ Waiting

містить три рядки, кожен з яких відповідає за визначену гілку подальшого руху. У даному випадку варіанти переміщення задаються імовірностями (0,3, 0,4 і 0,3 для пунктів призначення Auditor, Loan і Service відповідно). Можливо змінити значення імовірностей, але сума усіх значень повинна дорівнюватись 1. Час переміщення задається в полі Move Logic.

- процеси в пунктах Auditor, Loan і Service ідентичні - клієнт затримується на визначений час, нормально розгалужений в заданому інтервалі, а потім направляється до виходу (Depart).

- пункт Depart є кінцевим. Подальшим місцем проходження є EXIT - клієнти виходять із моделі.

Як виявилось, робоче місце Manager хоча і було оголошено, у моделі не використовується. Якщо робоче місце не використовується (не згадується в таблиці описів процесів), то на роботу моделі воно, відповідно, не впливає.

Все необхідне вже задано. Тепер можна запустити модель на виконання (Simulation|Run).



Рекомендації студентам. Під час роботи моделі зменшіть швидкість симуляції. Після роботи моделі перегляньте статистику, побудуйте різні види звітів (гістограми, графіки). Поверніться в модель, спробуйте змінити деякі параметри, наприклад, змінити частоту надходження клієнтів у модель або час чекання в деяких пунктах. Запустіть модель на виконання і порівняйте статистичні звіти.

Далі розглянемо модель **Phase2:**

Модель phase2. mod створена на основі phase1.mod і є її логічним продовженням та розширенням.

У цій моделі додана мережа переміщення клієнтів (Build|Path Networks), яка наведена на рисунку 4.13.

Path Networks							
Graphic	Name	Type	T/S	Paths...	Interfaces...	Mapping...	Nodes
	Client_Net	Passing	Speed & Dist	19	8	0	18

Paths			
From	To	BI	Distance
N1	N2	Bi	1.00
N2	N3	Bi	11.60
N2	N4	Bi	14.00
N4	N5	Bi	10.23
N4	N6	Bi	5.00
N6	N7	Bi	10.00
N7	N8	Bi	12.12
N6	N9	Bi	3.00

Рисунок 4.13 – Мережа переміщення клієнтів

При виклику даної функції у вікні Layout графічно відображається мережа переміщень.

У вікні Path Networks присутня мережа моделі Client_Net, котра містить у собі 19 шляхів (Paths) і 18 вузлів (Nodes).

Натиснувши на стовпці Graphic ви можете вибрати колір відображення шляхів у вікні Layout, а також зробити їх видимими або прихованими.

У полі Type даної мережі знаходиться значення Passing - тобто декілька клієнтів можуть одночасно переміщатися по одному шляху.

Вікно Paths відображає характеристики кожного шляху обраної мережі переміщень. Зокрема, між якими вузлами знаходиться шлях From(Від) To(До), довжину шляху (Distance), а також чи є шлях односпрямованим або двоспрямованим (UNI/BI).

Також в моделі присутній новий елемент – ресурс (рис.4.14), що моделює секретаря (Build|Resources).

Ви можете змінити графічне відображення ресурсу у вікні Resource Graphics, або модифікувати параметри ресурсу у вікні Resource.

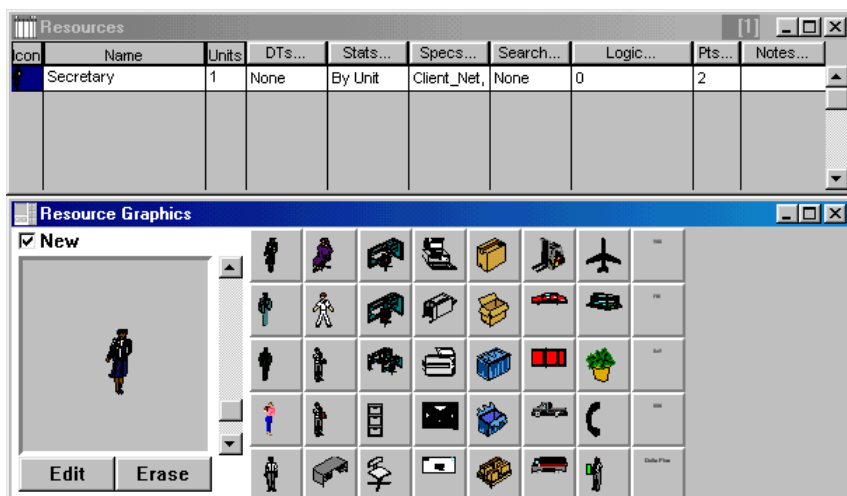


Рисунок 4.14 – Ресурс

Поля ресурсу у вікні містять його ім'я (Name), кількість представників даного ресурсу (Units), у полі DTs вибирається метод збору статистики ресурсом: None (немає), Basic (середня від усіх представників ресурсу), By Unit (своя для кожного представника).

Деякі поля викликають відповідний редактор для зміни більш глибоких параметрів ресурсу.

Для зручності спостереження за моделлю в Phase2, оголошені глобальні змінні (рис.4.15), що відслідковують кількість клієнтів у моделі на даний момент і кількість клієнтів, обслугованих за увесь час роботи (Build|More Elements|Variables).

Variables [global]					
Icon	ID	Type	Initial value	Stats...	Notes...
Yes	Clients_in_system	Integer	0	Time Series	
Yes	Clients_served	Integer	0	Time Series	

Рисунок 4.15 – Глобальні змінні

Усі відомості про глобальних змінні відображаються у вікні Variables: чи є графічне зображення у вікні Layout для даної змінної (Icon); ім'я, яке її ідентифікує (ID); тип (Type); початкове значення для змінної (Initial value), а також вид збираємої статистики (Stats).

Робоча область (вікно Layout) перетерпіла деякі зміни - додані лічильники, що прив'язані до глобальних змінних.

У зв'язку з присутністю нових елементів, більш витонченою стала логіка процесів на робочих місцях.



Рекомендації студентам. Пропонується простежити за роботою моделі, використовуючи приведені вище відомості.

Для розвитку власних навичок пропонується переглянути подальше удосконалення моделі у файлах phase3.mod, phase4.mod і phase5.mod., а також проаналізувати роботу інших стандартних моделей, що поставляються з CIM ServiceModel. Каталог models\refs містить невеликі моделі, що ілюструють ту або іншу команду або операцію. Імена файлів моделей як правило відповідають іменам команд.



Контрольні питання

1. За допомогою яких вікон і пунктів меню створюється робоче місце?
2. Як визначаються об'єкти обробки?
3. Як визначаються дії об'єктів?
4. Як створити вхідні потоки об'єктів обробки?
5. Назвіть види звітів в CIM ServiceModel.
6. За допомогою яких вікон графічно відображається мережа переміщень?
7. В якому вікні відображаються відомості про глобальних змінні?

4.3.4 Зарезервовані слова, вбудовані функції, процедури і команди, які використовуються при побудові логіки подій (*Logic Builder*)

ACCUM <Q>

Затримує об'єкти на визначеному робочому місці, доки в ньому не збереться Q об'єктів.

ACTIVATE <Sub> (Params)

Запускає незалежну підпрограму Sub (як правило визначену користувачем) з параметрами Params. Виконання команд продовжується, не чекаючи закінчення роботи підпрограми.

ALL

Зарезервоване слово ALL може використовуватися в якості ідентифікації об'єктів обробки всіх типів (якщо для усіх них передбачається та сама послідовність дій). Це ж слово може використовуватися в полі результату визначення маршруту, якщо для всіх типів об'єктів передбачається той самий маршрут. Також, можливо використання ALL у якості імені робочого місця у випадку, якщо усі вони асоціюються з тим самим процесом, а також мають однакову маршрутизацію.

ANIMATE <Speed>

Встановлює швидкість анімації моделювання ($1 < \text{Speed} < 100$). Чим вище значення, тим більше швидкість. По умовчання моделювання відбувається зі швидкістю 32.

ASSIGNMENT

Синтаксис: $\langle \text{ім'я} \rangle = \langle \text{значення} \rangle$

Присвоювання значення змінній, атрибуту або елементу масиву. При призначенні дрібного результату цілій змінній (атрибуту або елементу масиву) відкидається дрібна частина.

BEGIN

Початок операційного блока, що завершується оператором END (аналогічно операторним скобкам в Паскалі). Замість конструкції BEGIN-END можуть використовуватися фігурні скобки {}.

BETAB (<a>,,<c>,<d>) : real

Бетта-розподіл.

BI (<a>,) : real

Біноміальний розподіл, де a - розмір вибірки, а b - імовірність позитивного виходу.

BREAK

Припинення циклу і вихід з нього. Переривається найглибший (вкладений) WHILE-DO, DO-WHILE або DO-UNTIL цикл.

BREAKBLK

Вихід з операційного блока BEGIN-END.

CAP (<Loc>) : integer

Ємність робочого місця Loc.

CHAR (<Expr>)

Повертає один із 256 символів стандарту ANSI, код символу визначається виразом Expr.

CLOCK ([<T_unit>]) : real

Загальний час моделювання в одиницях T_unit (WK, DAY, HR, MIN або SEC). По умовчання - в одиницях часу, зазначених у вікні General Information (Загальна інформація).

CLOSE <FileID>

Закриває файл (ідентифікатор - FileID), із якого зчитувалася (READ) або в який записувалася (WRITE, WRITELINE, XWRITE) інформація.

COMBINE <Q> [AS <Ent>]

Збирає та об'єднує задане число (Q) об'єктів в один, обумовлений параметром Ent.

COMMENT

Вставка коментарів у блок логіки. Рядок також вважається коментарем, якщо в його початку знаходяться дві похилі риси '//'.
 //

CONTENTS (<Loc>[,<Ent>]) : integer

Поточне утримання місця обробки (загальне або по типу об'єктів Ent).

CREATE <Q> [AS <Ent>] [TAKE <Q1> <Res>]

Створення певної кількості Q нових об'єктів типу Ent, у додаток до існуючого. Всі нові об'єкти мають такі ж атрибути, як і їх "предок". Можливо також змусити ці об'єкти захопити Q1 одиниць ресурсу Res.

DEBUG

Виклик інтерактивного відлагоджувача. Під час роботи відлагоджувача відбувається виконання однієї команди за один крок. Також можливо переглядати значення змінних під час моделювання.

DEC <Ident> [,<Val>]

Зменшує значення змінної, елемента масиву або атрибута Ident на розмір Val (по умовчання - на одиницю).

DISPLAY <Msg>

Припиняє моделювання і виводить на екран вікно з повідомленням Msg. Моделювання відновляється при натисканні ОК.

DO <Statmnt> UNTIL <Cond>

Поки умова Cond ложна, циклічно виконується команда або операційний блок Statmnt.

DO <Statmnt> WHILE <Cond>

Поки умова Cond істинна, циклічно виконується команда або операційний блок Statmnt.

DOWNQTY (<Loc|Res>) : integer

Кількість одиниць робочого місця Loc або ресурсу Res, що знаходяться в неробочому стані.

DTDELAY ([<T_unit>]) : real

Різниця в часі між моментами, коли було заплановане встановлення непрацездатності, і коли воно в дійсності відбулося. Результат в одиницях T_unit (WK, DAY, HR, MIN або SEC).

DTLEFT ([<T_unit>]) : real

Час, що залишився, до кінця перерви (break) або до виходу на роботу ресурсу або місця обробки. Визначається відповідно до параметрів зміни, що задаються в редакторі змін (Shfit Editor). Результат в одиницях T_unit (WK, DAY, HR, MIN або SEC).

DTLENGTH () : real

Повертає тривалість перерви (break) або неробочого стану (off-shift). Визначається відповідно до параметрів зміни, що задаються в редакторі змін (Shfit Editor).

E (<a>) : real

Експоненціальний розподіл із середнім значенням a.

END

Закриває операційний блок, що починається оператором BEGIN.

ENTITY ([<N>]) : integer

Індекс імені N-го об'єкта в групі об'єктів. Якщо не задан параметр N, повертається індекс імені об'єкта, із якого визивається функція.

ENTRIES ([<Loc>]) : integer

Число входів у робоче місце Loc.

ER (<a>,) : real

Розподіл Ерланга з середнім експоненційним значенням a і параметром b.

EXP (<X>) : real

e (2,81...) у ступені X.

FORLOCATION () : integer

Повертає істину, якщо об'єкт, що працює в зміні або знаходиться на перерві, є місцем обробки.

FORMAT (<Expr>,<Front>[,<Rear>])

Дозволяє користувачу задавати формат відображення і виводу дійсних значень. Параметр Expr - чисельний вираз, що автоматично переводиться у формат дійсного числа. Front - число знаків до коми, Rear - число знаків після коми. FORMAT використовується тільки в рядкових виразах, тому перед ним необхідно завжди ставити значок \$.

FORRESOURCE () : integer

Повертає істину, якщо об'єкт, що працює в зміні або знаходиться на перерві, є ресурсом.

FREE [<Q1>] <Res1>, [<Q2>] <Res2>, ... [<Qn>] <Resn>

Змушує об'єкт звільнити Q1. .Qn одиниць ресурсів Res1. .Resn відповідно, якими він володіє. FREE ALL звільняє всі ресурси, якими володіє об'єкт на даний момент.

FREECAP (<Loc>) : integer

Вільна ємність робочого місця Loc.

FREEUNITS (<Loc|Res>) : integer

Число вільних одиниць місця обробки Loc або ресурсу Res.

G (<a>,) : real

Гамма-розподіл

GEO (<a>) : real

Геометричний розподіл, де a - імовірність успішного ісходу.

GET [<Q>] <Res> [, <Prio>] [, ...]

Робиться захоплення Q одиниць ресурса Res (або логічну комбінацію ресурсів, якщо заданий список) із пріоритетом Prio, як тільки ресурс звільняється.

GETCOST () : real

Повертає вартість поточного об'єкта в доларах.

GOTO <Label>

Здійснюється безумовний перехід до мітки Label. Використання конструкції If-THEN-ELSE більш переважно, ніж використання оператора GOTO.

GRAPHIC <GrNum>

Змінює графічне зображення поточного об'єкта на графічне зображення з номером GrNum.

GROUP <Q> [AS <Ent>]

Тимчасово групує позначену кількість (Q) об'єктів в один єдиний об'єкт типу Ent, що пізніше може бути розформований за допомогою команди UNGROUP.

GROUPQTY ([<Name>]) : integer

Кількість об'єктів певного типу в групі. Якщо параметр Name не заданий, повертається загальна кількість об'єктів у групі.

IF <Cond> THEN <YesAction> [ELSE <ElseAction>]

Типова конструкція ЯКЩО-ТО-ІНАКШЕ. Дія YesAction виконується, якщо умова Cond істинна, інакше виконується дія ElseAction.

IG (<a>,) : real

Розподіл, інверсний Гауссовському.

INC <Ident> [, <Val>]

Збільшує значення змінної, елемента масиву або атрибута Ident на указаний розмір Val (або на 1 по умовчання).

INCENTCOST <Val>

Підвищує вартість поточного об'єкта на розмір Val.

INCLOCCOST <Val>

Підвищує вартість поточного об'єкта на розмір Val.

INF

Нескінченно великий розмір.

INT <Name> [= <Val>]

Створює локальну змінну цілого типу і привласнює їй значення. (Локальні змінні доступні тільки всередині логічного блока, у якому вони визначені).

JOIN <Q> <Ent> [, <Prio>]

Приєднує певну кількість (Q) об'єктів типу Ent до поточного об'єкта з пріоритетом Prio.

JOINTLY GET [<Q>] <Res> [, <Prio>] [, ...]

Захоплення Q одиниць ресурсу Res (або декількох ресурсів, якщо заданий логічний список) із пріоритетом Prio. Захоплення здійснюється тільки коли ВСІ ресурси будуть вільні, на відміну від GET, де ресурси захоплюються в міру їхнього звільнення.

L (<a>,) : real

Логнормальний розподіл.

LAST () : integer

Індекс назви попереднього вузла, у якому знаходився ресурс.

LN (<X>) : real

Натуральний логарифм від X.

LOAD <Q> [IFF <Cond>] [IN <Time>] [, <Prio>]

Тимчасово приєднує задану кількість об'єктів (Q, або $\leq Q$ протягом заданого періоду Time) до поточного об'єкта з пріоритетом Prio, якщо дотримується умова Cond. Відбувається не групування об'єктів, а своєрідне завантаження одного об'єкта іншими.

LOCATION () : integer

Індекс імені поточного місця обробки.

LOG <Str>, <Attr_ID>

Віднімає час, збережений у змінній, атрибуті або масиві Attr_ID із системного часу (часу моделювання) і записує значення разом із рядком Str у файл для наступного перегляду в Модулі вихідних даних (Output Module).

MATCH <Attr>

Змушує систему дочекатися, поки значення атрибута Attr поточного об'єкта не збіжиться зі значенням цього ж атрибута іншого об'єкта системи. Логіка іншого об'єкта також повинна містити відповідний вираз MATCH.

MOVE [FOR <Time>]

Переміщає об'єкт до кінця конвеєра або черги перед робочим місцем протягом часу Time. Має пріоритет вище, ніж параметр “швидкість об'єкта” (див. Параметри Entities).

MOVE ON <Net>

Примушує об'єкт рухатися вздовж маршрутної мережі Net.

MOVE WITH <Res1> [OR <Res2>] [FOR <Time>] [THEN FREE]

Змушує об'єкт робити рух, використовуючи (захопивши) ресурс Res1 (або альтернативно Res2) протягом часу Time. Якщо вираз завершується командою THEN FREE, то після руху відбувається звільнення ресурсу.

N (<a>,) : real

Нормальний розподіл з середнім значенням a і середньоквадратичним відхиленням b.

NEXT () : integer

Індекс назви вузла, у який надалі буде рухатись ресурс.

ORDER <Q> <Ent> TO <Loc>

Створюється Q об'єктів типу Ent. Об'єкти вводяться в систему в робочому місці Loc. Якщо ємність робочого місця недостатня, об'єкти які не помістилися удаляються.

OWNEDRESOURCE ([<N>]) : integer

Індекс назви N-го ресурсу, яким володіє об'єкт на даний момент. Параметр N визначає порядковий номер ресурсу в хронологічному списку доступних ресурсів. По умовчання повертається останній захоплений ресурс, яким володіє об'єкт.

P (<a>) : real

Пуассонівський розподіл з середнім значенням а.

P5 (<a>,) : real

P6 (<a>,,<c>) : real

Розподіли Пірсона (Пірсон5, Пірсон6).

PAUSE

Призупинення моделювання. Моделювання відновляється при виборі опції Resume із меню Simulation.

PRIORITY <Prio>

Зміна пріоритету робочого місця, що простоює, або ресурсу. Якщо пріоритет менше попереднього значення, система робить перевірку, чи може робоче місце (ресурс) бути розвантажено.

PROCESSNUM () : integer

Номер процесу, що викликається (часто використовується при відлагодженні).

PROMPT <Stringexpr>, <Name> [,<Choice1>:<Val1>, <Choice2>:<Val2> ...]

Моделювання припиняється і робиться запит на введення значення змінної, атрибута або елемента масиву. Спливаюче вікно містить безпосередньо запит системи (Stringexpr), а також поле введення значення, що призначається змінній (атрибуту, елементу масиву) Name. Якщо список можливих значень задається у вигляді <Опис>:<Значення>, то користувачу пропонується вибрати одне з них.

RAND (<X>) : real

Випадкове значення з інтервалу $0 \leq N < X$.

READ (<Filename>,<Name>)

Считує наступне значення з файла Filename і призначає його змінній, атрибуту або елементу масиву Name.

REAL <Name> [= <Val>]

Створює локальну змінну дійсного типу та привласнює їй значення. (Локальні змінні доступні тільки всередині логічного блока, у якому вони визначені).

RENAME AS <Ent>

Призначення нового імені (типу) Ent поточному об'єкту.

REPORT [WITH RESET] [AS <Name>]

Підраховує і записує поточну статистику у вихідну базу даних.

RESOURCE ([<N>]) : integer

Індекс назви N-го ресурсу, яким володіє об'єкт на даний момент. По умовчанню повертається останній захоплений ресурс, яким володіє об'єкт.

RESQTY ([<Res>]) : integer

Кількість одиниць певного ресурсу Res, що належить поточному об'єкту. Якщо не позначене ім'я ресурсу, то повертається загальна кількість одиниць ресурсів усіх типів.

RETURN [<Val>]

Припиняє підпрограму і повертає значення Val блоку, який викликав. При використанні поза підпрограмами діє як потужний оператор BREAK, завершуючи не тільки операційний блок, але і всю нитку логіки.

ROUND (<X>) : integer

Округлення X до цілого по правилах округлення.

ROUTE <Block>

Змушує об'єкт іти по певному блоку маршрутизації Block процесу.

SEND <Q> <Ent> TO <Loc> [,<Prio>]

Робиться запит на створення (посилання) Q нових об'єктів типу Ent на робочому місці Loc із пріоритетом Prio.

SKIP

У предпочатковій логіці (pre-start logic) змушує об'єкт (робоче місце) пропустити перерви і простої. У основній логіці (main logic) ігнорує певний в редакторі змін час перерви або час простою.

SOUND <Filename>

Програє звуковий . wav файл Filename.

SPLIT <Q> AS <Ent>

Розщеплює поточний об'єкт на Q об'єктів типу Ent. Всі значення атрибутів зберігаються.

SQRT (<X>) : real

Квадратний корінь від X .

STOP <Msg>

Зупиняє моделювання і виводить на екран повідомлення Msg.

T (<a>,,<c>) : real

Трикутний розподіл із мінімальним значенням a , максимальним b і модою c .

TIMESUSED (<Res>) : integer

Показує, скільки разів використовувався ресурс Res.

TRUNC (<X>) : integer

Приведення до цілого типу шляхом відкидання дрібної частини.

U (<a>,) : real

Постійний розподіл із середнім значенням a і половиною ширини інтервалу b .

UNGROUP [LIFO]

Розгрупує об'єкти, об'єднані оператором GROUP. Якщо не указан параметр LIFO, процедура виконується по методу FIFO.

UNITS (<Loc|Res>) : integer

Число одиниць місць обробки (Loc) або ресурсів (Res), що знаходяться в системі.

UNLOAD <Q> [IFF <Cond>]

Примушує об'єкт, навантажений іншими об'єктами в результаті команд GROUP або LOAD, "вивантажити" із себе Q об'єктів, якщо виконується умова Cond.

USE [<Q>] <Res> [,<Prio>] FOR <Time>

Захоплення поточним об'єктом Q одиниць ресурсу Res (або комбінації ресурсів) відповідно до пріоритету, як тільки ресурс звільниться. Ресурс захоплюється на час, визначений параметром Time.

W (<a>,) : real

Розподіл Вейбулла.

WAIT <Time>

Затримка об'єкта на час Time.

WAIT UNTIL <Cond>

Затримка об'єкта доти, поки не буде виконуватися умова Cond.

WHILE <Cond> DO <Stmnt>

Циклічна структура. Дія або операторний блок Stmnt виконується доти, поки буде виконуватися умова Cond.

WRITE (<Fileid>, <Str> | <Val> [,<Front>,<Rear>])

Запис чисельного значення Val або символьного рядка Str у загальний файл виходу, визначений у редакторі файлів (File editor). Чисельне значення виводиться з Front знаками перед комою і Rear знаками після неї.

WRITELINE

Те ж, що і WRITE, але після запису відбувається повернення каретки.

XSUB (<Params>)

Виклик зовнішньої процедури, визначеної в .dll файлі з параметрами Params.

XWRITE <File>, <Str>

Запис символьного рядка Str у файл File.

6 ПРИКЛАДИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ

Приклад 1. Сервер обслуговує дві локальні підмережі А и В великої організації з 8:00 до 19:00. У комп'ютерів підмережі А виникає необхідність використовувати комунікації зв'язку одразу ж після включення серверу порціями по 64Кб кожні 2 хвилини та тривалістю 2с.

У комп'ютерів підмережі В виникає необхідність використовувати мережеві з'єднання через 30 хвилин після включення сервера порціями по 512 Кб щохвилини та тривалістю 10с.

Провести моделювання системи на протязі 10 годин з урахуванням того, що обслуговування підмереж сервером здійснюється послідовно і через 2 години після включення сталося відключення електрики на 7 хвилин.

Приклад 2. Роботу центрального процесора комп'ютерної системи (КС) можуть переривати сигнали від пристроїв трьох типів: системного годинника, диска і терміналів. Процесор обслуговує ці переривання в залежності від пріоритетів. Найвищий пріоритет у системного годинника, потім слідує диск та термінал.

Сигнали від системного годинника поступають 18 раз за секунду одразу ж після увімкнення КС. У КС встановлено два диска. Переривання від першого надходять кожні 5 хв, через 20 хвилин після увімкнення комп'ютера, а від другого – кожні 8 хв одразу ж після включення. Переривання від терміналу поступають кожні 20 секунд, через 2 хвилини після ввімкнення.

Провести моделювання роботи КС на протязі години (з 8-00 до 9-00) з урахуванням того, що з 8-20 до 8-30 була відключена електрична енергія.

ЛІТЕРАТУРА

Основна література

1. Системный анализ в экономике и организации производства / Валуев С.А. и др. -Л.: Политехника, 1991.-398с.
2. Советов Б.Я., Яковлев С.А./ Б.Я. Советов, С.А.Яковлев. Моделирование систем. -М.: ВШ, 1985. -271с.
3. Кузин Л.Т. Основы кибернетики.Т.2 Основы кибернетических моделей / Л.Т. Кузин. -М.: Энергия, 1979. -584с.
4. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженеров / О.П. Кузнецов, Г.М.Адельсон-Вельский. - М.: Энергаториздат, 1988. -480с.
5. Максимей И.В. Имитационное моделирование на ЭВМ / И.В. Максимей. -М.: Радио и связь, 1988. -232с.
6. Тимченко А.А. Основи проектування та системного аналізу складних об'єктів: Основи системного підходу та системного аналізу об'єктів нової техніки / А.А. Тимченко. -К.: Либідь, 2004.- 288с.

Додаткова література

7. Губанов В.А., Захаров В.В. Введение в системный анализ / В.А. Губанов, В.В. Захаров. -Л.: ЛГУ, 1988. -232с.
8. Губинский А.И. Надежность и качество функционирования эргатических систем / А.И. Губинский. -Л.: Наука, 1982. -270с.
9. Шрайбер Т.Дж. Моделирование на GPSS / Т.Дж. Шрайбер - М.: Машиностроение, 1980. -592с.
10. Попович П.Р., Губинский А.И., Колесников А.М. Эргономическое обеспечение деятельности космонавтов / П.Р. Попович, А.И. Губинский, А.М. Колесников. -М.: Машиностроение, 1985. -272с.
11. Советов Б.Я., Яковлев С.А. Моделирование систем: Учебник для вузов / Б.Я. Советов, С.А. Яковлев. – М.: ВШ, 2005. – 343 с.
- 12 Кельтон В., Лоу А. Имитационное моделирование / В. Кельтон, А. Лоу – СПб: Питер; Киев: Изд. группа BHV, 2004. - 847 с.