**УДК 519.6:519.179**

*Гайтан О.М., старший викладач,*

*Ткаленко І.О., Масич О.С., студенти,*

*Полтавський національний технічний університет*

*імені Юрія Кондратюка*

**ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНИХ СЕРЕДОВИЩ МОДЕЛЮВАННЯ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ**

**Вступ**

У процесі розробки інформаційних і обчислювальних систем для оцінки їх продуктивності, розподілу ресурсів тощо активно використовуються системи імітаційного моделювання.

Імітаційне моделювання представляє собою метод дослідження, при якому система, що досліджується, замінюється моделлю, що з достатньою точністю описує реальну систему, для проведення експериментів з моделлю з метою одержання інформації про цю систему. Методи імітаційного моделювання широко використовуються для дослідження складних систем у різних сферах людської діяльності завдяки зниженню вартості використання комп’ютерної техніки та наявності спеціалізованих програмних засобів.

Особливо ефективне моделювання при вирішенні наступних завдань:

* проектування та аналіз виробничих систем;
* оцінка різних систем озброєнь;
* визначення вимог до устаткування та протоколів мереж зв'язку;
* модернізація різних процесів у діловій сфері;
* аналіз фінансових і економічних систем.

Значне поширення отримали системи імітаційного моделювання для дослідження систем масового обслуговування (СМО).

Перші дослідження СМО виконані у 20-х роках минулого століття A. Ерлангом для підвищен­ня якості обслуговування телефонних ліній.

На сьогоднішній день на основі теорії масового обслуговування побудовано багато моделей реальних систем. Зазвичай за допомогою методів теорії масового обслуговування розв'язують задачі з проектування та експлуатації однотипних елементів обслуговування, наприклад, розраховують кількість контрольно-пропускного обладнання, місць для ремонту, бензоколонок, обслуговуючого персоналу, ліній зв'язку, одиниць обладнання обчислювальної техніки тощо.

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

**Формулювання мети статті.** Метою статті є порівняльний аналіз функці­ональних можливостей сучасних програмних середовищ для моделю­вання систем масового обслуговування, розгляд їхніх переваг та недоліків, а також надання рекомендацій з їх вибору.

**Виклад основного матеріалу**

**1. Поняття про системи масового обслуговування.**

Під системою масового обслуговування розуміється об’єкт, що складається з обслуговуючої системи та системи, що обслуговується разом з правилами, які встановлюють організацію обслуговувань (рис. 1.).



Рис. 1. Загальна схема СМО (a, b, …, s – об’єкти обслуговування або джере­ла вимог; 1, 2, ..., n – обслуговуючі пристрої або канали обслуговування)

Основними елементами систем масового обслуговування є:

* вхідний потік вимог;
* черга;
* канали обслуговування;
* вихідний потік вимог (обслужені вимоги).

Структурна схема типової системи масового обслуговування наведена на рис. 2.



Рис.2. Структурна схема типової системи масового обслуговування

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

* середня кількість вимог, які система може обслужити за одиницю часу;
* середній відсоток вимог, які не були обслужені;
* ймовірність того, що вимогу буде прийнято для обслуговування;
* середній час очікування вимоги у черзі;
* закон розподілу часу очікування;
* середня кількість вимог у черзі;
* коефіцієнт завантаження пристрою для обслуговування;
* середня кількість пристроїв, зайнятих обслуговуванням.

Для визначення цих параметрів необхідно описати та задати такі харак­теристики СМО:

* вхідний потік вимог;
* дисципліни поставки вимог у чергу та вибору вимог із неї;
* правила, за якими здійснюється обслуговування;
* вихідний потік вимог;
* режими роботи системи.

**2. Програмні середовища моделювання систем масового обслугову­вання.**

Імітаційне моделювання СМО можна виконувати як за допомогою мов загального призначення, за і за допомогою спеціалізованих прикладних пакетів. Класифікація інструментальних засобів наведена на рис. 3.

Fortran

Java

C#

C++

Extend

AnyLogic

Arena

GPSS

Системи моделювання СМО масового обслуовування

Спеціалізовані системи

Мови програмування загального призначення

Рис. 3. Класифікація інструментальних засобів моделювання СМО

Використання алгоритмічних мо­в загального призначення дає змогу реалізовувати прості та більш швидкодіючі імітаційні моделі. Але у такому випадку з метою уніфікації програмних модулів і принципів структурного та модульного програмування необхідно вручну роз­робити засоби:

* керування процесом моделювання в модельному часі;
* збирання статистичних даних про роботу моделі;
* генерування випадкових величин з різними законами розподілів ймовірностей для моделювання стохастичних впливів;
* діагностики помилок під час виконання програми моделювання.

Розглянуті засоби існують у будь-якому спеціалізованому пакеті прикладних про­грам, орієнтованому на моделювання. Тому у разі реалізації складної імітаційної моделі, що має велику кількість різних компонентів, перевагу належить віддавати спеціалізованим засобам моделюван­ня.

Одними з найбільш популярних пакетів імітаційного моделювання СМО на сьогоднішній день є:

1. GPSS World.
2. Arena компанії.
3. AnyLogic
4. Extend

При виборі засобів імітаційного моделювання варто враховувати всі можливості, що вони надають, які можна об'єднати в наступні групи: основні характеристики; сумісне програмне забезпечення; анімація; ста­тистичні можливості; звіти з вихідними даними і графіками; документація.

**3. Загальна характеристика пакетів імітаційного моделювання СМО.**

**3.1. *GPSS***

***GPSS*** (англ. General Purpose Simulation System – система моделювання за­галь­ного призначення) – мова моделювання, розроблена співробітником фірми IBM Джефрі Гордоном в 1961 році, яка використовується для імі­та­цій­но­го моделювання різних систем, в основному СМО систем масового обслуговування. Система GPSS розроблена для професіоналів в області моделювання, досить проста у вивченні й універсальна в застосуванні.

Динамічним елементом моделі в GPSS є транзакт – абстрактний об'єкт, що переміщується між статичними елементами, відтворюючи різні події реального модельованого об'єкта. В процесі роботи моделі нако­пи­чується статистика, яка автоматично виводиться по завершенні процесу мо­делювання. Статичні елементи моделі: джерела транзактів, пристрої, черги тощо. Їх розташування в моделі визначається блоками.

Приклад програми у програмному середовищі GPSS:

\* Определение входных данных

 RMULT 413

Zapas STORAGE 2000 ; Вместимость склада.

Zapas TABLE Zapas,0,10,2000 ; Гистограмма запаса.

Material VARIABLE Nach\_ur-S$Zapas ; Размер заказа.

Spros\_m VARIABLE RN1@15+35 ; Размер ежедневного спроса.

Nach\_ur EQU 1500 ; Начальный уровень запаса.

\*\*\*\*\*\*\*\*\*

GENERATE 200,,,,11

TERMINATE 1

\* Моделирование процесса поставки материала на склад.

 GENERATE 5,,,,1 ; Поставка через 5 дней.

 TEST L S$Zapas,1000,Outt ; Нужна ли поставка.

 ASSIGN 2,V$Material ; Размер заказа в Р2.

 ENTER Zapas,P2 ; Увеличение запаса на Р2.

Outt TERMINATE ; Завершение заказа.

\*\*\*\*\*\*\*\*\*

\* Моделирование текущего запаса с учетом спроса.

 GENERATE 1 ; Ежедневный спрос.

 ASSIGN 1,V$Spros\_m ; Размер спроса в Р1.

 TABULATE Zapas ; Запись текущего запаса.

 TEST GE S$Zapas,P1,Zapasout ; Можно ли заказать.

 LEAVE Zapas,P1 ; Уменьшение запаса на P1.

Zapasout TERMINATE ; Завершение дня.

\*\*\*\*\*\*\*\*\*

\*Установление начального размера запаса.

 GENERATE ,,,1,10 ; Начальный запас.

 ENTER Zapas,Nach\_ur ; Установка начального запаса.

 TERMINATE ; Завершение установки.

**3.2. Arena**

***Arena*** – програмне забезпечення для імітаційного моделювання, розроблене компанією Rockwell Automatіon, яке дозволяє будувати рухомі комп’ю­терні моделі, використовуючи які можна адекватно уявити багато реальних систем.

Моделі з допомогою Arena можуть бути побудовані для таких сфер діяльності як: виробничі технологічні операції, складський облік, банківська діяльність, обслуговування клієнтів у ресторані тощо.

Arena дозволяє:

* моделювати процеси на подальше дослідження, документування;
* візуалізацію процесів за допомогою динамічної графіки і мультиплікації;
* розробляти процеси, що дозволяють боротися з невизначеністю, викликаною випадковістю і непостійністю системи.

В системі використовується процесор і мова SІMAN. Arena надає користувачеві графічний інтерфейс із набором шаблонів моделюючих кон­струк­цій. Спроектована під потреби почат­ків­ців, але до­сить потужна, щоб задо­вольнити вимоги досвідчених користу­вачів, сис­тема Arena адаптується під рівень досвідченості користувача.

Приклад моделі у програмному середовищі Arena наведений на рис. 4:



Рис. 4. Приклад моделі у програмному середовищі Arena

**3.3 AnyLogic**

AnyLogic – програмне забезпечення для імітаційного моделювання бізнес-процесів, розроблене російською компанією The AnyLogic Company в 2000 р.

Моделі AnyLogic можуть ґрунтуватися бути засновані на будь-якій з основних парадигм імітаційного моделювання: дискретно-подійного моделю­вання, системної динаміки, і агентного моделювання.

AnyLogic дозволяє відображати динаміку економічних і соціальних систем на різних рівнях абстракції. AnyLogic включає набір примітивів і бібліотечних об'єктів для моделювання виробництва і логістики, бізнес-процесів і персоналу, фінансів, споживчого ринку, а також навколишньої інфраструктури в їх природній взаємодії. Об'єктно-орієнтований підхід, пропонований AnyLogic, полегшує ітеративну поетапну побудову великих моделей.

У редакторі можна розробити анімацію та інтерактивний графічний інтерфейс моделі. У AnyLogic включені засоби аналізу даних і набір елементів бізнес-графіки, спроектованих для обробки та презентації результатів моделювання: статистики, набори даних, графіки, діаграми, гістограми.

Програма підтримує різні типи експериментів з моделями: простий прогін, порівняння прогонів, варіювання параметрів, Монте-Карло, аналіз чутливості, оптимізація, калібрування, а також довільний експеримент по призначеному для користувача сценарієм.

Мовою для опису структур даних, дій, правил і алгортімов в AnyLogic є Java.

Приклад моделі у програмному середовищі AnyLogic наведений на рис. 5:



Рис. 5. Приклад моделі у програмному середовищі AnyLogic

***3.4. Extend***

***Extend*** – це програмне середовище для дискретного моделювання подій, розроблене Imagine Trat Inc. За допомогою Extend досить просто моделювати складні системи з часовою залежністю.

Extend володіє можливостями документування і надає графічний інтерфейс для візуальної побудови, спостереження та аналізу імітаційних моделей дискретних подій. У Extend підтримується проста анімація, яка дозволяє відображати 40 переміщення клієнтів або завдань в системі, яка моделюється.

Пакет імітаційного моделювання Extend забезпечує автоматичне виконання різних сценаріїв, підтримуваних системою з використанням еволюційного оптимізатора з відкритим кодом.

Також наявні інтерактивний налагоджувач початкових кодів, побудова моделей реалізована на основі компонентів ActiveX, користувачу надається навігатор в стилі Провідника, можливість обміну даними через Web інтерфейс, виконання кількох моделей одночасно.

Приклад моделі у програмному середовищі Arena наведений на рис. 6:



Рис. 6. Приклад моделі у програмному середовищі ***Extend***

***3.5. Порівняльний аналіз середовищ моделювання систем.***

Порівняльна характеристика наведених вище програмних середовищ моделювання СМО наведена у табл. 1.

Таблиця 1. Порівняльна характеристика наведених вище програмних середовищ моделювання СМО

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Критерій** | **GPSS** | **Arena** | **AnyLogic** | **Extend** |
|  | Сфера застосування | Універсальна мова моделю­вання дис­крет­них сис­тем, в т.ч. СМО. | Виробництво, лан­цюги поста­вок/ логістика, управ­ління біз­нес процесами, війсь­ково-промисло­вий комплекс, медицина. | Менеджмент,ви­­робництво, обслуговування,логістика, лан­цюги поста­вок, меди­цина, транспорт, IT, уп­равління, те­ле­комунікації, наука. | СМО, вклю­ча­ючи збутову ло­гістику, call-центри з ве­ли­ким наван­та­жен­ням, паку­вальні лінії. |
|  | Візуальне середовище | − | + | + | + |
|  | Елементи СМО / системи | Черга, при­стрій, накопи­чу­вач, тран­закт.  | Черга, пристрій, сутність. Бло­ки, коректо­ри. | Залежить від ви­­браної па­ра­дигми моде­лю­вання: сис­тем­­на ди­на­міка; ди­ск­­ретно-подійне або агентне мо­де­­лю­вання. | Блоки (дії), стріл­ки (шляхи переміщення); ром­­­би (розга­лу­ження шля­хів); черги; точ­ки вво­ду дина­міч­них об'єктів у модель. |
|  | Бібіліотеки |  |  | Process Modeling;Pedestrian;Rail Yard |  |
|  | Графічні можливостіПобудова графіків Анімація  | Побудова графіків, гістограм, графічні вікна для спосте­реження за процесом моделювання. | Об’єктно-орієнто­ваний інтерфейс, анімаційна система Cinema animation. | Вбудована ані­ма­ція і ство­рен­ня аплетів, оп­ти­мізатор Opt­Quest, швид­кий механізм пла­ну­вання експе­ри­мен­тів. | Інтегрований еволюційний оптимізатор, інтерактивний відладчик ви­хідних кодів, COM/ ActiveX, навігатор, діа­грами Ганта. |
|  | Можливість програмування | PLUS | Simfn | Java | MODL |
|  | Можливість відладки | + | + | + | + |
|  | Генерування звітів | + | + | + | + |
|  | Системні вимоги | 64MB min., 128MB recom. Windows 95, 98, ME, NT, 2000 and XP | 64MB min., 128MB recom. Windows 95, 98, ME, NT, 2000 and XP  | 128MB min., 512MB recom. Windows NT, 2000 and XP  | 64MB min., 128MB+ recom.Windows XP, 2000, NT, ME, 98; Power Macintosh OS X and 9.1+ |

За допомогою вищевказаних програмних систем було виконане імі­та­ційне моделювання продовольчого супермаркету з декількома від­ді­ла­ми та касами з заданими законами розподілу часу приходу та обслу­го­ву­вання по­куп­ців. Згідно з побудованою моделлю були визначені: кое­фі­­ці­єнти заван­та­­ження кожного касира та продавця-консультанта; макси­маль­не, середнє і поточне число покупців у кожній черзі; середній час обслу­го­ву­вання на кожній касі і в кожному відділі; середній час перебування по­купця в кож­ній черзі тощо. За результатами моделювання була визначена оптимальна кількість кас, щоб максимальна довжина черги не переви­щувала 5 чоловік.

Результати моделювання, отримані в різних системах, майже іден­тич­ні, але досягнути точного повторення результатів неможливо через те, що різні програмні середовища використовують різні методи округлення чисел з ко­мою, що плаває, різні генератори випадкових чисел тощо.

**Висновки**

 За резуль­та­тами імітаційного моделювання можна зробити наступні висновки:

1. Найбільшу функціональність (3 вбудовані бібліотеки для різних галузей застосування, підтримка різних парадигм моделювання, різні види графіки та діаграм тощо) має AnyLogic, проте даний програмний продукт є платним. Крім того, використання даного продукту вимагає певного часу для ознайомлення та освоєння.
2. Arena не вимагає написання програмного коду і проста у вико­ристан­­ні, але має обмеження на 150 сутностей для некомерційної версії продукту. Стандартна відкрита архітектура Arena дозволяє створювати власні інтерфейси та додатки. При використан­ні на підприємстві про­фесійний випуск Arena здатний забезпечити йому платформу розробки для створення власних комплектів адаптованих інструментів моделювання.
3. Завдяки візуальному середовищу (Arena, AnyLogic тощо) можна спостерігати перебіг моделювання в реальному часу, задавати анімаційні картинки сутностям або ресурсам, що значно покращує візуальне сприйняття результатів.
4. На відміну від системи візуальних середовищ GPSS World є досить «прозорим» (струк­тура моделі не є чорним ящиком), що допускає нестандартну оброб­ку даних; але необхідно відзначити наявність у неї ряду недо­ліків, зокрема повільна робота інтерпретатора та необхідність напи­сан­ня програмного коду.
5. Наявність вбудованих в середовища алгоритмічних мов програмування дозволяє розширювати можливості стандартних засобів та створювати власні надбудови. Так, наприклад, алгоритмічна мова PLUS в GPSS значно розширила можливості системи порівняно зі стандартною мовою GPSS і дала змогу включити в систему засоби планування проведення експериментів. А можливість програмування мовою Java для AnyLogic дозволяє сторювати власні аплети при розробці моделей.
6. Використання будь-якого програмного середовища для імітаційного моделювання все ж вимагає від користувача досить глибоких знань тео­рії ймовірностей, математичної статистики і теорії систем масо­во­го обслу­го­вування.
7. Кожний програмний продукт орієнтований на розв’язання певного класу задач, тому при виборі конкретного засобу треба орієнтуватися на поставлені задачі та вимоги, що висуваються до результатів моделювання.

***Література***

1. *Томашевський, В. М. Моделювання систем / В. М. Томашевський. – К.: BHV, 2005. – 352 c.*
2. *Томашевский, В. Н. Имитационное моделирование в среде GPSS / В. Н.Томашевский, Е. Г. Жданова. – М.: Бестселлер, 2003. – 416 с.*
3. *Боев, В. Д. Моделирование систем. Инструментальные средства GPSS World / В. Д. Боев. – СПб.: БХВ-Петербург, 2004. – 368 с.*
4. *Кудрявцев, Е. М. GPSS World. Основы имитационного моделирования различных систем / Е. М. Кудрявцев. – М.: ДМК Пресс, 2004. – 320 с.*
5. *Лега Ю. Г. Методи імітаційного моделювання систем та процесів: практикум : навч. посіб. / Ю. Г. Лега, А. Д. Кожухівський, О. А. Кожухівська. – Черкас. держ. технол. ун-т. – Черкаси: ЧДТУ, 2010. – 247 с.А*

*Гайтан О.М., старший викладач,*

*Ткаленко І.О., Масич О.С., студенти,*

*Полтавський національний технічний університет*

*імені Юрія Кондратюка*

**ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНИХ СЕРЕДОВИЩ МОДЕЛЮВАННЯ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ**

*У статті проведено порівняльний аналіз функці­ональних можливостей сучасних програмних середовищ для моделю­вання систем масового обслуговування, розглянуті їхні переваги та недоліки, а також надані рекомендації з їх вибору.*

***Ключові слова****: візуалізація, моделювання, черга, система масового обслуговування, транзакт.*

*Гайтан Е.Н., старший преподаватель,*

*Ткаленко И.А., Масич А.С., студенты,*

*Полтавский национальный технический университет*

*имени Юрия Кондратюка*

**СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ПРОГРАММНЫХ СРЕД МОДЕЛИРОВАНИЯ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ**

*В статье проведен сравнительный анализ функциональных возможностей современных программных сред для моделирования систем массового обслуживания, рассмотрены их преимущества и недостатки, а также предоставлены рекомендации по их выбору.*

***Ключевые слова****: визуализация, моделирование, очередь, система массового обслуживания, транзакт.*

*Gajtan E.N., Senior Lecturer,*

*Tkalenko I.O., Masich A.S., students,*

*Poltava National Technical University*

*named after Yuri Kondratyuk*

**THE COMPARATIVE CHARACTERISTICS OF SOFTWARE
FOR QUEUING SYSTEMS SIMULATION**

*The article provides a comparative analysis of the features of modern software systems for queuing systems simulation, their* [*benefits and drawbacks*](http://www.lingvo.ua/ru/Search/Translate/GlossaryItemExtraInfo?text=%d0%bf%d1%80%d0%b5%d0%b8%d0%bc%d1%83%d1%89%d0%b5%d1%81%d1%82%d0%b2%d0%b0%20%d0%b8%20%d0%bd%d0%b5%d0%b4%d0%be%d1%81%d1%82%d0%b0%d1%82%d0%ba%d0%b8&translation=benefits%20and%20drawbacks&srcLang=ru&destLang=en)*, and recommendations on making choice are given.*

***Keywords****: visualization, simulation, queue, queuing system, transact.*