

КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

*O. Pletinka, N. Brovarska
A. Potapenko*

THE PROBLEM OF ADAPTATION POLICY ENVIRONMENT FOR VIRTUAL SIMULATION OF SCIENTIFIC EXPERIMENT

*In the robot explores the issues of
adaption and harmonization of soft-
ware tools for modeling virtualnogo
scientific experiment.*

*Key words: virtual simulation, virtu-
al environment, methodology, pro-
gramming languages.*

*Исследуются вопросы адаптации
и унификации программных сред
для виртуального моделирования
научного эксперимента.*

*Ключевые слова: виртуальное
моделирование, виртуальное ок-
ружение, методология, языки
программирования.*

*Досліджуються питання адапта-
ції та уніфікації програмних сере-
довищ для віртуального моделю-
вання наукового експерименту.*

*Ключові слова: віртуальне моде-
лювання, віртуальне оточення,
методологія, мови програмування.*

© О.В. Плетінка, Н.Й. Броварська,
А.Ф. Потапенко, 2013

УДК 681.3.06

О.В. ПЛЕТІНКА, Н.Й. БРОВАРСЬКА, А.Ф. ПОТАПЕНКО

ДО ПИТАННЯ АДАПТАЦІЇ ПРОГРАМНИХ СЕРЕДОВИЩ ДЛЯ ВІРТУАЛЬНОГО МОДЕЛЮВАННЯ НАУКОВОГО ЕКСПЕРИМЕНТУ

Вступ. Побудова систем віртуального моде-
лювання (СВМ) з різноманітними формами
віртуального оточення (ВО) є одним з най-
більш потужних засобів моделювання склад-
них об'єктів для процесу організації та про-
ведення наукових експериментів. Розвиток
напряму, що пов'язаний з створенням вір-
туальних прототипів (ВП), допомагає вирі-
шувати різноманітні практичні задачі не
звертаючись при цьому до фізичного анало-
гу. Віртуальні прототипи дозволяють відмо-
витися від натурних моделей і проводити
аналіз всіх параметрів на етапах проектуван-
ня, тестування, підготовки та проведення
автоматизованого експерименту. Водночас,
завдяки своїй близькості до фізичної моделі,
організація систем віртуального оточення
дозволяє у повній мірі провести оцінки варі-
антів отримання необхідних характеристик
для кожного етапу життєвого циклу дослі-
джень.

Як відомо [1], формальне подання систем
віртуальної групи ґрунтується на моделю-
ванні реально існуючих об'єктів і об'єктів, що
підлягають створенню у майбутньому. Мо-
делювання таких об'єктів має загальний ме-
тодологічний підхід, але істотно відрізняєть-
ся в частині прикладних методів досліджен-
ня. При цьому, доцільно використовувати
концепцію моделювання, в якій формально-
методичну платформу складають математич-
ні моделі, орієнтовані на наукові дисципліни,
що вивчають ті предметні області, які підля-
гають дослідженню.

У цьому випадку, математична модель не

є самоціллю, а лише засобом для вирішення певної проблеми [2]. У зв'язку з цим необхідність створення математичної моделі впливає з обраної дослідником методології постановки і вирішення проблеми. Для складних проблем зазвичай застосовують системний підхід, в якому моделювання є основним методом дослідження. У цілому такий підхід передбачає наступні етапи вирішення проблеми:

- аналіз предметної області;
- виявлення і формулювання проблеми;
- формалізована постановка проблеми;
- аналіз фізичних і математичних параметрів визначених процесів і об'єктів;
- підбір засобів для моделювання ВП досліджуваних процесів і об'єктів;
- формування підходів, вимог і вибір середовища моделювання ВО для ВП;
- моделювання ВП;
- обробка результатів моделювання;
- оцінка альтернативних рішень;
- формулювання висновків і пропозицій щодо вирішення проблеми.

Загальна частина. Для побудови та вирішення задач віртуального моделювання системи ВО можуть бути подані у вигляді діалектичної єдності апаратних засобів, програмного забезпечення та інформаційно-методологічно забезпечення. Структуру кожного забезпечення можливо подати різноманітними функціональними класами, які з точки зору користувача зводяться до двох основних аспектів: забезпечувати максимальну зручність та забезпечувати максимальну ефективну організацію віртуального процесу для цільового призначення.

Для роботи процесу віртуального моделювання однією з проблем є використання ефективних мов програмування, які в певній мірі визначають якість програмного продукту та чіткість роботи системи в цілому. Вибір мови програмування пов'язаний з багатьма факторами, які відображають віртуальну задачу, прикладну галузь, рівень віртуального оточення та глибину наукових досліджень. Тому вибір мови програмування має відповідати різноманітним показникам до яких відносяться такі властивості: надійність як відповідність програмного продукту своєму призначенню, невеликі затрати праці на створення програмного забезпечення, мала трудоемкість використаних програм, а також показники, які характеризують такі властивості: продуктивність програмування, можливість адаптації розроблених програмних засобів до зміни віртуальної задачі та віртуального оточення з урахуванням моделювання різноманітних мультимедійних компонент.

Тому задача адаптації програмних середовищ для віртуального моделювання має велике значення. Сьогодні не існує спеціалізованих мов програмування, які б дозволяли в повній мірі будувати віртуальне оточення для моделювання різних класів досліджень. З цього приводу, питання адаптації мов та середовищ програмування для задач побудови систем віртуального оточення є вкрай важливим. Розробка загальних підходів та уніфікованої методології для різних мов програмування дозволить створити єдиний уніфікований підхід для побудов систем віртуального моделювання незалежно від галузі.

Розглянемо існуючі програмні середовища, які можуть бути використані для рішення задач ВО.

Одним з перспективніших середовищ для створення віртуального оточення є програмний комплекс Аванго [3], який використовує мову програмування C++ для визначення двох категорій об'єктів – вузлів та датчиків. Вузли складають об'єктно-орієнтовані граф-сцени, які й здійснюють подання і відображення складної геометрії. Датчики забезпечують зв'язок з реальним світом і використовуються, щоб вводити дані із зовнішніх пристроїв у додаток. Всі об'єкти Аванго – польові контейнери, що представляють інформацію про стан об'єкта як сукупність полів. Об'єкти Аванго підтримують універсальний потоковий інтерфейс, який дозволяє записувати в потік об'єкти та інформацію про їх стан, і згодом реконструювати ці дані з потоку. Аванго визначає зв'язки між полями, формуючи граф потоків даних, який є концептуально ортогональним до графа сцени і служить для введення в сцену реальних даних, необхідних для моделювання інтерактивної поведінки. Крім C++ Аванго підтримує інтерпретовану мову Схема (Scheme) [4]. Це мова високого рівня, яка вміє оперувати зі структурними даними, такими як рядки, списки і вектори. Всі об'єкти Аванго можуть бути описані на Схемі. Аванго використовує графічну систему OpenGL Performer [5], яка дозволяє досягти максимально можливого прискорення графіки і також володіє рядом спеціальних властивостей, необхідних при розробці додатків VE. Розширені завдання, такі як стирання невидимих граней, перемикання ступеня деталізації і зв'язок з апаратним забезпеченням, цілком виконуються системою OpenGL Performer.

Програмне забезпечення всієї системи в цілому будується на базі ОС Linux засобами інструментальної системи віртуального оточення Аванго і мови C++. Розроблюваний комплекс програм має забезпечувати платформно-незалежне функціонування на основі технології Java [6]. Комплекс перевіряється на наступних операційних системах: Linux, Windows NT, XP, Solaris. Як апаратні платформи використовуються високопродуктивні персональні комп'ютери і робочі станції. Як СУБД може використовуватися будь-яка система, що підтримує драйвер JDBC. При розробці використовуватися інструментальні засоби RAD JBuilder, а також CASE-технології, інтегровані з технологіями JAVA (разом JB). Для розробки програм застосовується сучасна мова моделювання UML [6].

Базова вимога до мов та середовищ програмування – універсальність. Не всі пакети та програмні рішення відповідають критерію універсальності. Пакети, орієнтовані на вузькі прикладні області (електроніка, електромеханіка і т. д.) не є універсальними, оскільки елементи складних систем відносяться, як правило, до різних прикладних областей. Серед решти пакетів, що позиціонують себе як універсальні (орієнтованих на певну математичну модель), не відповідають вимогам побудови ВС та ВО пакети, орієнтовані на математичні моделі, відмінні від простої динамічної системи (рівняння часткових похідних, статистичні моделі), а також на чисто дискретні і чисто безперервні. Таким чином, предметом розгляду будуть універсальні пакети побудови систем віртуального оточення, що дозволяють моделювати структурно-складні гібридні системи.

Якщо розглядати ряд уніфікованих пакетних рішень для побудови моделей систем віртуального оточення, можна умовно поділити їх на три групи:

- 1) пакет "блокового моделювання";
- 2) пакети "фізичного моделювання";
- 3) пакети, що орієнтовані на схему гібридного автомата.

Цей поділ є умовним насамперед тому, що всі ці пакети мають багато спільного: дозволяють будувати багаторівневі ієрархічні функціональні схеми, підтримують у тій чи іншій мірі технологію об'єктно-орієнтованого моделювання, надають подібні можливості візуалізації та анімації. Відмінності обумовлені тим, який з аспектів складної динамічної системи є найбільш важливим.

Пакети "блокового моделювання" орієнтовані на графічну мову ієрархічних блок-схем. Елементарні блоки є або функціонально визначеними, або можуть конструюватися за допомогою деякої спеціальної допоміжної мови більш низького рівня. Ці мови програмування є стандартизованими або взяті з основи мов високого рівня: C++, Java, .Net. Новий блок можна зібрати з наявних блоків з використанням орієнтованих зв'язків та параметричних налаштувань. У число зумовлених елементарних блоків входять безперервні, дискретні і гібридні блоки.

До переваг цього підходу слід віднести, перш за все, надзвичайну простоту створення моделей. Іншою перевагою є ефективність реалізації елементарних блоків і простота побудови еквівалентної системи. Водночас, при створенні складних моделей доводиться будувати досить громіздкі багаторівневі блок-схеми, що не відображають природної структури модельованої системи. Іншими словами, цей підхід працює добре, коли є відповідні стандартні блоки. Найбільш відомими представниками пакетами "блокового моделювання" є [7]:

- підсистема SIMULINK пакета MATLAB (MathWorks, Inc.);
- EASYS (Boeing);
- SystemDuild MATRIXX (Integrated Systems, Inc.);
- VisSim (Visual Solution).

Пакети "фізичного моделювання" дозволяють використовувати неорієнтовані і потокові зв'язки. Користувач може сам визначати нові класи блоків системи. Безперервна складова поведінки елементарного блоку задається системою алгебро-диференціальних рівнянь і формул. Дискретна складова задається описом дискретних подій (події задаються логічною умовою чи є періодичними), при виникненні яких можуть виконуватися миттєві привласнення змінним нових значень. Дискретні події можуть поширюватися за спеціальними зв'язками. Зміна структури рівнянь можливо тільки побічно через коефіцієнти в блоках управління (це обумовлено необхідністю символічних перетворень при переході до еквівалентної системи).

Підхід дуже зручний і природний для опису типових блоків фізичних систем. Недоліками є необхідність символічних перетворень, що різко звужує можливість опису гібридної поведінки, а також необхідність чисельного рішення великого числа алгебраїчних рівнянь, що значно ускладнює завдання автоматичного отримання достовірного рішення. Також більшість цих пакетів дозволяє

візуалізувати дані моделювання, підключати високорівневі мови програмування для вирішення складних задач.

До пакетів "фізичного моделювання" слід віднести [8]:

- "20-SIM" (Controllab Products B.V.);
- Dymola (Dynasim);
- Omola, OmSim (Lund University).

Як узагальнення досвіду для розвитку систем цього напрямку міжнародна група вчених розробила мову Modelica (The Modelica Design Group), запропоновану міжнародною науковою спільнотою як стандарт при обміні описами моделей між різними пакетами. Modelica є об'єктно-орієнтованою мовою моделювання для компонентно-орієнтованих складних систем, наприклад, систем, що містять механічні, електричні, електронні, гідравлічні, теплові, контролюючі, електроенергетичні або процесно-орієнтовані підчастини. Асоціація ентузіастів розвиває й розробляє безкоштовні бібліотеки для різних галузей. Сьогодні існує більше ніж 1280 загальних компонентів моделей і 910 функцій у різних областях, починаючи з версії 3.2, Modelica є міжнародним трансгалузевим стандартом при розробці систем віртуального оточення.

Пакети та мови програмування, засновані на використанні схеми гібридного автомата, дозволяють дуже наочно і природно описувати гібридні системи зі складною логікою взаємозв'язків. Необхідність визначення еквівалентної системи при кожному перемиканні змушує використовувати тільки орієнтовані зв'язки. Користувач може сам визначати нові класи блоків. Безперервна складова поведінки елементарного блоку задається системою алгебро-диференціальних рівнянь і формул. До недоліків слід також віднести надмірність опису при моделюванні чисто безперервних систем.

До цього напрямку відноситься пакет Shift (California PATH), а також вітчизняний пакет Model Vision Studium (MVS). Пакет Shift більше орієнтований на опис складних динамічних структур, а пакет MVS – на опис складної поведінки. Зауважимо, що між другим і третім напрямками немає непереборної прірви. Зрештою, складність спільного використання обумовлена лише сьогоdnішніми обчислювальними можливостями. Водночас, загальна ідеологія побудови систем віртуального оточення практично збігається. В принципі, можливий комбінований підхід, коли в структурі моделі мають виділятися складові блоки, елементи яких мають чисто безперервну поведінку, і одноразово перетворюються до елементарного еквіваленту. Далі вже сукупна поведінка цього еквівалентного блоку має використовуватися при аналізі гібридної системи.

Велике значення при реалізації моделі в ВО на ЕОМ має питання правильного вибору мови програмування. Мова програмування має відображати внутрішню структуру при описі широкого кола понять. Високий рівень мови моделювання значно спрощує програмування систем віртуального оточення.

Основними моментами при виборі програмних мов та середовищ моделювання є:

- проблемна орієнтація;
- можливості збору, обробки, виведення результатів;

- швидкодія;
- простота налагодження;
- доступність сприйняття.

Цими властивостями володіють процедурні мови високого рівня. Для моделювання можуть бути використані мови імітаційного моделювання (МІМ) [8] і мови загального призначення (МЗП). Останні є універсальними, але мають ряд обмежень, які в МІМ вже вирішені. Не дивлячись на наявність ряду готових бібліотек та функціональних блоків під МЗП для побудови систем віртуального оточення, рішення галузевих задач вимагає більш детальної кастомізації програм, що суттєво ускладнює адаптацію МЗМ для задач моделювання та побудови систем віртуального оточення. Більш зручними, звичайно, є МІ. Вони забезпечують вирішення основних задач, так як в повній мірі розроблені для задач створення систем віртуального оточення та моделювання:

- зручність програмування моделі системи;
- проблемна орієнтація.

Але разом з тим, МІМ мають й недоліки:

- неефективність робочих програм;
- складність налагодження;
- відсутність повної документації.

Основні функції МІМ програмування:

- управління процесами (узгодження системного і машинного часу);
- управління ресурсами (вибір і розподіл обмежених коштів системи).

Як спеціалізовані мови, МІМ володіють деякими програмними властивостями і поняттями, які не зустрічаються в МЗП. До них відносяться:

- поєднання. Паралельно протікаючи в реальних системах **S** процеси представляються на ЕОМ за допомогою послідовних процесів. МІМ дозволяють обійти ці труднощі шляхом введення понять системного часу;

- розмір. МІМ використовують динамічний розподіл пам'яті (компоненти моделі системи **M** з'являються в ОЗУ і зникають залежно від поточного стану). Ефективність моделювання досягається при використанні блокових конструкцій (блоків, підблоків і т. д.);

- зміни. МІМ передбачають обробку списків, що відображають зміни станів процесу функціонування моделюючої системи на системному рівні;

- взаємозв'язок. Для відображення великої кількості зв'язків між компонентами моделі в статистиці і динаміці, МІМ використовують системно організовані логічні можливості та реалізацію теорії множин;

- стохастичність. МІМ використовують спеціальні програмні генерації послідовностей випадкових чисел, програми перетворення у відповідні закони розподілу;

- аналіз. МІМ передбачають системні способи статистичної обробки та аналізу результатів моделювання.

Найбільш відомі мови моделювання – SIMULA, SIMSCRIPT, GPSS, SOL, CSL.

Представлення системи S у вигляді типової схеми, в якій беруть участь як дискретні, так і безперервні величини, називаються комбінованими. Передбачається, що в системі можуть відбуватися події двох видів:

- 1) події, від стану Z_i ;
- 2) події, що залежать від часу T .

При використанні мови GAPS на користувача покладається робота по складанню на мові FORTRAN підпрограм, в яких описуються умови настання подій, закони зміни безперервної величини, правила переходу з одного стану в інший.

SIMSCRIPT – мова подій, створена на базі мови FORTRAN. Кожна модель M складається з елементів, з якими відбуваються події, що представляє собою послідовність формул, які змінюють стан модельованої системи з впливом часу; працюють зі списками, що обумовлені користувачем, створюють послідовність подій у системному часі та забезпечують роботу з множинами.

FORSIT – пакет ПП на мові FORTRAN дозволяє оперувати тільки фіксованими масивами даних, що описують об'єкти які моделюються. Пакет зручний для опису систем з великим числом різноманітних ресурсів. Повний опис динаміки моделі можна отримати за допомогою ПП.

SIMULA – розширення мови Алгол. Моделюється блочне подання. Функціонування процесу розбивається на етапи, що відбуваються в системному часі. Головна роль у мові SIMULA відводиться поняттю паралельного оперування процесами в системному часі, універсальній обробці списків з процесами в ролі компонент.

GPSS – інтегруюча мовна система, що застосовується для опису просторового руху об'єктів. Такі динамічні об'єкти в мові GPSS називаються транзактами і являють собою елементи потоку. Транзакти "створюються" і "знищуються". Функцію кожного з них можна представити як рух через модель M з почерговим впливом на її блоки. Функціональний апарат мови утворюють блоки, що описують логіку моделі, повідомляючи транзактам, куди рухатися і що робити далі. Дані для ЕОМ готуються у вигляді пакету керуючих і визначальних карток, які складаються за схемою моделі, набраної з стандартних символів. Створена програма GPSS, працюючи в режимі інтерпретації, генерує і передає транзакти з блоку в блок. Кожен перехід транзакта приписується до певного моменту системного часу.

При моделюванні, перевагу віддають мовам, які більш універсальні. Але разом із збільшенням числа команд зростають труднощі використання МІМ для розробки міждисциплінарних систем віртуального оточення.

Класифікація мов, що використовуються для побудови систем віртуального оточення показана на рисунку.

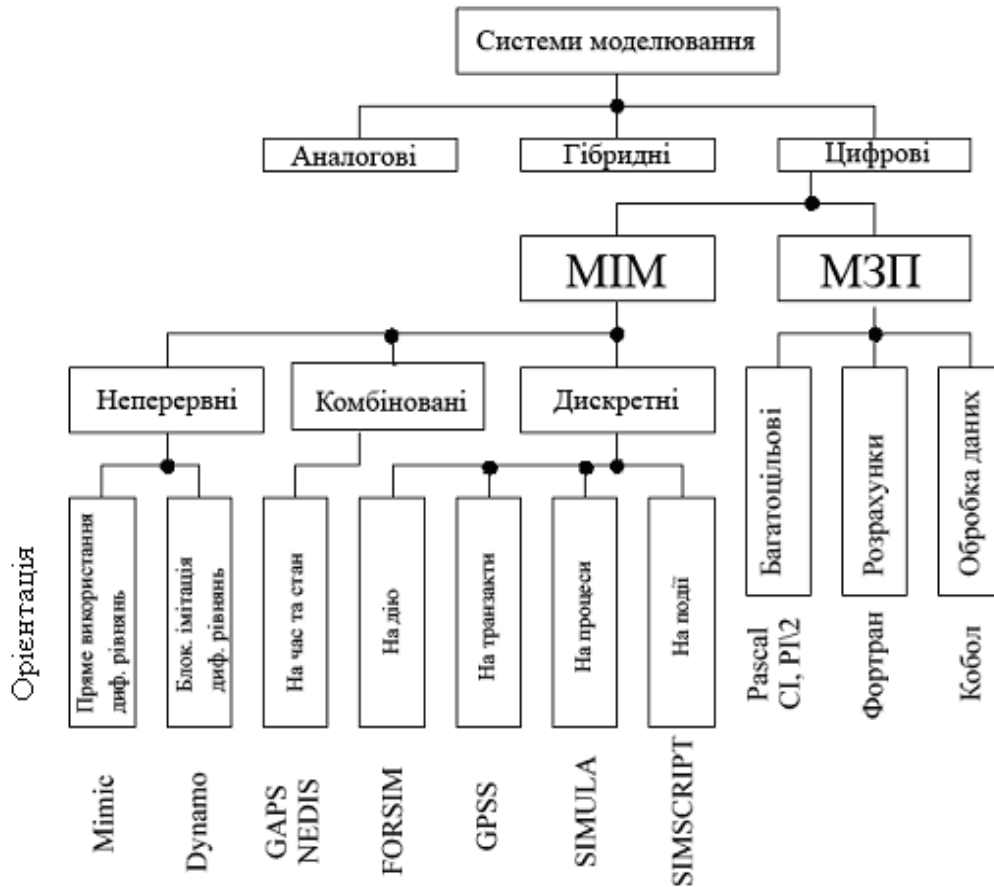


РИСУНОК. Класифікація мов, що використовуються для побудови систем віртуального оточення

Висновки. Методологія побудови систем віртуального оточення висуває ряд вимог до програмних мов та середовищ, та ґрунтується на моделюванні реально існуючих об'єктів і об'єктів, що підлягають створенню у майбутньому. Моделювання таких об'єктів має загальний методологічний підхід, але істотно відрізняється в частині прикладних методів дослідження. При цьому, доцільно використовувати концепцію моделювання, в якій формально-методична платформа базується на адаптації програмних середовищ з ціллю уніфікації методів побудови систем віртуального оточення. Для цього необхідно розробити ряд вимог-критеріїв до програмних середовищ. Дотримання цих критеріїв, при виборі мови програмування, гарантуватиме адаптивність, універсальність, гнучкість та можливість візуалізації. Це забезпечить ефективний вибір мов програ-

мування, пакетів та середовищ та суттєво спростить використання мов програмування й дозволить уніфікувати адаптацію програмних середовищ для вирішення задач побудови віртуального оточення для різних галузей науки та промисловості. Уніфікація й поява єдиних підходів для вибору мов програмування та програмних середовищ ВО дозволить проводити міжгалузеві та міждисциплінарні дослідження різними групами розробників та дослідників одночасно.

1. *Корсаков А.М.* Исследование возможностей построения 3D моделей внешней среды и отдельных объектов многоакурсной системы технического зрения // Искусственный интеллект. Интеллектуальные и многопроцессорные системы. Материалы междунар. науч.-техн. конф. Таганрог – Донецк, – 2004. – Том. 2. – С. 411 – 414.
2. *Устенко А.С.* Основы математического моделирования и алгоритмизации процессов функционирования сложных систем. – СПб: Издательство Бином, “Невский диалект”, 1998. – 560 с.
3. *Никитина Л.Д.* Электронный журнал «Исследовано в России», с. 552, Фраунгоферовский Институт Медиакоммуникаций, Санкт Августин, Германия.
4. *Kent Dybvig R.* The Scheme programming language: ANSI Scheme. P T R Prentice-Hall, Englewood Cliffs, NJ 07632, USA, second edition, 199.
5. *Никитина Л.Д.* Электронный журнал «Исследовано в России», с. 553, Фраунгоферовский Институт Медиакоммуникаций, Санкт Августин, Германия.
6. *Алешин В.В.* Создание технологии виртуального окружения для решения проблем распознавания космических аппаратов и оценки их параметров по земным наблюдениям. – Институт физико-технической информатики, Протвино – Москва, 2005.
7. *Колесов Ю.Б., Сениченков Ю.Б.* Имитационное моделирование сложных динамических систем [Элект. ресурс] (http://www.exponenta.ru/soft/others/mvs/ds_sim.asp).
8. *Воронин А.В.* Моделирование мехатронных систем. – Издательство Томского политехнического университета, 2007. – С. 106.

Одержано 12.10.2013