

УДК 004.94

Б. Б. Нестеренко, М. А. Новотарський

Інститут математики НАН України

вул. Терещенківська, 3, 01601 ГСП, Київ-4, Україна

тел. (044) 234-04-07, e-mail: model@imath.kiev.ua, novot@ukr.net

Програмний комплекс для моделювання складних систем на основі алгебри процесів

Обґрунтовано необхідність застосування алгебри процесів як формального засобу для опису складних систем. Розглянуто основи побудови програмного комплексу, що базується на розширеній алгебрі процесів. Описано принципи організації препроцесорної мови для алгебри процесів та підсистеми моделювання. Проведено порівняння запропонованого програмного комплексу з відомими середовищами моделювання подібного типу.

Ключові слова: імітаційне моделювання, алгебра процесів, операція, процес, активність.

Вступ

Сучасні вимоги до аналізу складних дискретних систем вимагають застосування якісно нових підходів до створення моделей. Більшість згаданих систем характеризуються складною структурою компонентів і нелінійним характером взаємодії між ними. Тому однією з актуальних задач залишається збереження заданого рівня взаємної подібності між об'єктом моделювання та його моделлю. Прагнення одержати ефективний розв'язок цієї задачі спонукає розробників до створення більш детальних моделей, які змогли б максимально врахувати особливості складної системи. Це, у свою чергу, зумовлює нові труднощі, що проявляються в стрімкому зростанні громіздкості моделі та неможливості гарантування коректності одержаних результатів. Одним з актуальних підходів до вирішення цієї проблеми є розробка нових і модифікація відомих формальних засобів для опису об'єкта моделювання з метою забезпечення заданого рівня абстрагування без утрати адекватності.

Останні десять років стали періодом бурхливого розвитку формальних засобів опису складних систем. Вони пройшли шлях від скінченних автоматів [1] та агрегатів [2], орієнтованих на послідовний характер моделювання, до паралельних мережних засобів, що беруть свій початок від мереж Петрі [3]. Однак оптимізм щодо застосування мережних засобів формального опису почав зникати через ряд

© Б. Б. Нестеренко, М. А. Новотарський

проблем, які виникли при спробах забезпечити строге доведення подібності об'єкта дослідження до його моделі шляхом побудови дерева досяжності [4]. Крім того, структурна наочність мережних моделей обернулася громіздкістю при описі складних систем, що істотно ускладнило розробку інтерфейсів програмних комплексів, які базуються на мережних формальних засобах.

Ідея нових засобів формального опису складних систем полягає в максимальному наближенні їх до математичних виразів, що дозволяє створювати компактні інтерфейси програмних комплексів, які включають вхідну мову моделювання, а також застосовувати інструментарій строгого доведення фактів взаємної подібності, базуючись на основних положеннях загальної теорії систем. Саме такими властивостями характеризуються алгебри процесів.

Представлений у даній роботі програмний комплекс базується на спеціально створеній версії алгебри процесів [5], яка забезпечує моделювання складних систем із використанням реального робочого навантаження. Він складається з інтерфейсної підсистеми та підсистеми імітаційного моделювання. Інтерфейсна підсистема забезпечує введення, редагування та збереження опису моделі, що задана за допомогою високорівневої препроцесорної мови моделювання.

Принципи організації препроцесорної мови для алгебри процесів (ПМАП)

Основною вимогою до розроблюваної мови є наявність у ній операторів, що дозволяють описувати моделі у виразах алгебри процесів. Також мова забезпечує просте й зручне задавання реального робочого навантаження для створеної моделі та усуває ряд синтаксичних недоліків алгебри процесів [6].

Важливою особливістю підсистеми лексичного та синтаксичного аналізу програмного комплексу є те, що вона базується на використанні об'єктів алгебри процесів: активностей і процесів. Над цими об'єктами визначені спеціальні оператори й операції. При визначенні процесу можна задати ідентифікатор області змінних. Кожна активність, що зв'язана з конкретним класом (реальне робоче навантаження), має можливість зберігати певні змінні в єдиній області змінних і розділяти їх з іншими активностями. Ідентифікатор області змінних указує, з якою саме областю буде працювати активність. Активності з різних областей змінних не можуть обмінюватися інформацією через розділювані змінні.

У мові визначені два оператори алгебри процесів. Оператор присвоєння алгебри процесів (=) використовується для асоціації певного імені зі складеним процесом, тобто для задавання складеного процесу на основі більш простих. Оператор задавання реального робочого навантаження (define) забезпечує зв'язок імені зовнішнього класу з активністю.

Розглянемо коротко принципи реалізації основних операцій алгебри процесів.

Операція префіксації (.). Процеси, що об'єднані операцією префіксації, виконуються строго послідовно один за одним. Жодна подія наступного процесу не може бути запланована в системі, якщо є хоча б одна запланована подія попереднього процесу. У випадку, коли попередній процес є складним, наступний процес очікує завершення обходження дерева цього процесу.

Операція паралельної композиції з асинхронною взаємодією (||). Процеси, що об'єднані операцією паралельної композиції, розвиваються паралельно, тобто планування подій у цих процесах виконується незалежно. Вони можуть взаємодіяти шляхом виконання активностей взаємодії. При цьому блокування процесу-передавача не відбувається, а всі наступні події можуть плануватись й виконуватись незалежно від того, чи відбулась в системі комплементарна подія взаємодії. Операція моделює взаємодію паралельних компонентів системи. Асинхронна взаємодія дозволяє моделювати, наприклад, роботу мережних протоколів із заданим рівнем абстракції.

Операція паралельної композиції із синхронною взаємодією (|). Аналогічно до попередньої операції, процеси, що беруть участь у цій операції, розвиваються паралельно, і їхні обчислювальні події плануються й виконуються незалежно одна від одної. Однак виконання активностей взаємодії синхронізується з виконанням комплементарної їй активності. Це означає, що після планування активності взаємодії жодна подія не може бути виконана або запланована в цьому процесі до тих пір, поки не буде виконана комплементарна подія взаємодії.

Операція вибору (+). Ця операція дозволяє виконання тільки одного процесу із числа тих, що беруть у ній участь. Решта процесів завершуються, і жодна подія для них більше не планується.

Спосіб реалізації вибору може бути достатньо складним. Коли настає момент вибору, обирається процес, який має найбільше значення «пріоритету». Якщо із процесом асоційований клас робочого навантаження, «пріоритет» процесу визначається спеціальною функцією в цьому класі, що дозволяє здійснювати розгалуження процесу виконання моделі на рівні робочого навантаження. Якщо ж із процесом не асоційований жодний клас, пріоритет процесу обирається випадковим чином із рівномірного розподілу від 0 до 1. Одночасно можуть здійснюватись операції вибору в паралельних гілках процесу та на різних рівнях ієрархії. Для коректного розв'язання цих операцій вибору кожен процес, що задіяний в операції, одержує свою мітку, яка унікальна для кожної операції вибору. Після вибору одного із процесів решта процесів з аналогічною міткою знищуються. Цей процес не зачіпає решту операцій вибору із власними мітками.

Операція приховування ($\{L\}$). Операція приховування є важливою операцією розбиття моделі на логічні блоки. Фактично дана операція обмежує область видимості активностей, що перераховані в множині L , тим процесом, за яким іде дана операція. Жодна активність, що відбувається «всередині» операції, не може бути помічена «назовні». Це означає, що комплементарні активності синхронної взаємодії, які знаходяться в різних «областях видимості», не можуть бути використані для взаємодії двох процесів. Взаємодія може відбутись тільки всередині відповідної операції приховування.

Опис об'єкта моделювання, що заданий на мові ПМАП, підлягає перетворенню в послідовність виразів алгебри процесів, які безпосередньо можуть бути прийняті підсистемою імітаційного моделювання.

Підсистема імітаційного моделювання

Моделювання складних систем здійснюється підсистемою імітаційного моде-

лювання на основі подій (рис. 1).

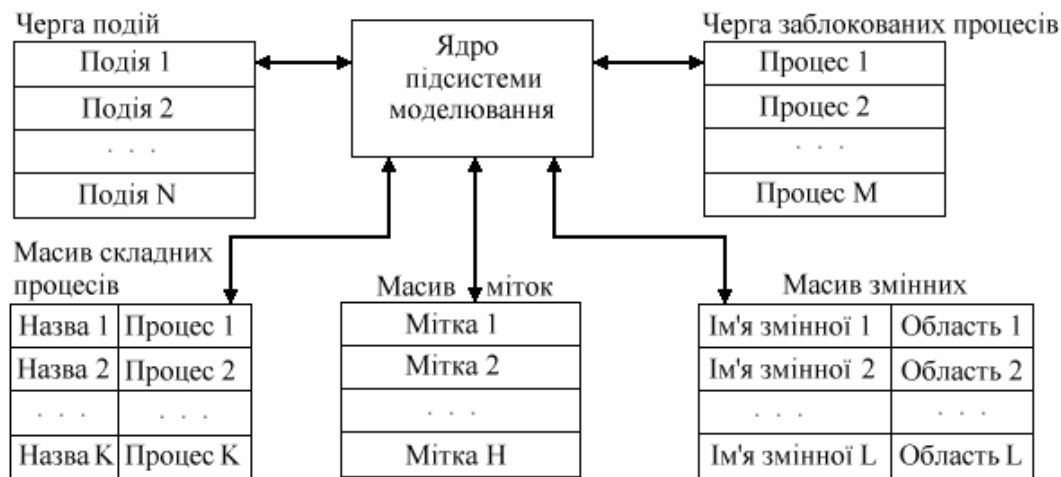


Рис. 1. Структура підсистеми імітаційного моделювання

При такому способі моделювання основним елементом підсистеми є черга подій, що формується ядром підсистеми моделювання у відповідності до часових міток цих подій. Функціонування моделі полягає у виборі подій із черги й занесенні нових подій у чергу.

Черга заблокованих процесів включає процеси, які перейшли в заблокований стан при виконанні операції синхронної взаємодії. Процеси знаходяться в заблокованому стані до моменту здійснення іншим процесом комплементарної операції взаємодії.

Асоціативний масив складних процесів також входить до складу підсистеми моделювання і є результатом роботи програми попереднього задавання моделі. У цьому масиві зберігаються відповідності між іменем процесів і деревом підпроцесів, з яких складаються ці процеси. Асоціативний масив областей змінних — структура, що може містити змінні, якими обмінюються активності. До складу підсистеми імітаційного моделювання також входить масив міток, що містить мітки тих операцій вибору, які на даний момент моделювання ще не розв'язані.

Моделювання починається із планування події, що відповідає кореневому процесу SYSTEM, занесення події в чергу подій, а також дублювання дерева процесів, що асоційоване із процесом SYSTEM. Джерелом для дублювання є асоціативний масив складних процесів, що складається з певного набору шаблонів, з яких у ході моделювання створюються необхідні дерева. Вказівник на результат дублювання зберігається у відповідному полі підсистеми моделювання.

Розглянемо основні типи подій, що виникають у системі.

Початок обчислювальної активності — подія, яка планується для кожного елемента складних процесів. При виконанні цієї події відбувається перевірка списку складних процесів. Якщо ім'я обчислювальної події знайдене в списку складних процесів, то наступною планується подія «початок складного процесу». Час звершення цієї події дорівнює поточному модельному часу. Фактично, це є пер-

шим кроком розгортання складного процесу в ієрархію його компонентів. Якщо ж процесу з таким ім'ям не знайдено, то відбувається планування події «кінець обчислювальної активності». Час планування події залежить від декількох факторів. Перед визначенням часу відбувається перегляд асоціацій між подіями й класами реального робочого навантаження. Якщо з подією не асоційовано жодного класу, то час виконання активності береться випадковим чином з експоненціального розподілу з параметром розподілу, рівним одиниці. Якщо з подією асоційований клас реального робочого навантаження, то час планування події визначається спеціальною функцією цього класу.

Кінець обчислювальної активності задають подією, яка символізує завершення активності, і є ключовою подією в системі, оскільки саме вона просуває модельний час, а також реалізує обробку реального робочого навантаження. При виникненні цієї події модельний час просувається до часової мітки події. Після цього відбувається перевірка, чи асоційований з активністю клас реального робочого навантаження. За відсутності такого класу обробка події завершується. Якщо ж необхідний клас існує, то здійснюється виконання його основної функції. Функція класу містить набір інструкцій на мові програмування Java і є робочим навантаженням моделі. При дослідженні обчислювальних алгоритмів, наприклад, функція може містити одну ітерацію цього алгоритму. Таким чином, у ході моделювання складної системи відбувається реальне виконання її програмних елементів, що дозволяє розглядати модель не лише як елемент дослідження, але і як реальний обчислювальний процес, що має свою мету, здійснює обчислення, може вводити й виводити дані. Отже, модель у вигляді виразів алгебри процесів розглядають як додатковий шар над обчисленнями, який керує обчислювальним процесом на макрорівні, й веде певну статистику для оцінки ефективності тієї чи іншої реалізації системи, що моделюється.

Незалежно від того, чи асоційований із подією клас реального робочого навантаження, здійснюється планування наступної події. Тип і спосіб планування залежить від процесу, складовою частиною якого є поточна активність.

Початок складного процесу супроводжується створенням технічної події, яка відображає заміну назви процесу його деревом підпроцесів. При обробці цієї події в дереві моделі відбувається заміна листового елемента (що містив назву процесу) відповідним деревом, яке дублюється з масиву складних процесів. У цей час також відбувається планування подій відповідно до операції в корені дерева.

Кінець складного процесу також супроводжується технічною подією, яка символізує завершення моделювання даного складного процесу. При її обробці знищується дерево складного процесу (з метою економії пам'яті), і планується подія відповідно до операції на вищому кроці ієрархії.

Початок активності взаємодії — подія, яка відображає початок здійснення активності комунікації. У залежності від типу взаємодії розглянемо два варіанти розвитку подій. У випадку асинхронної комунікації обробка події майже нічим не відрізняється від обробки обчислювальної події. Тобто відбувається перевірка наявності класу робочого навантаження, планується подія «завершення активності взаємодії». Якщо ж взаємодія синхронна, то відбувається перегляд процесів, що знаходяться в черзі заблокованих процесів, з метою пошуку процесу, який очікує

взаємодії з даним процесом (тобто процесу з комплементарною активністю взаємодії).

У випадку, коли такий процес знайдено, він «розблоковується» — тобто планується подія «завершення активності взаємодії», і вилучається із черги заблокованих процесів. Для поточної активності також планується подія «завершення активності взаємодії». Обидві події плануються на один і той же модельний час. Час, як і для обчислювальних подій, визначається класом робочого навантаження.

Якщо пошук у черзі заблокованих процесів дав негативний результат, то процес блокується сам, тобто розміщується в черзі заблокованих процесів, і жодна наступна подія не планується.

Кінець активності взаємодії — подія, що символізує завершення активності взаємодії між двома процесами. Аналогічно до подій завершення обчислювальних активностей у цій активності виконується функція класу робочого навантаження, що безпосередньо здійснює передачу даних від одного процесу до іншого. Після виконання цих функцій плануються наступні події, що залежать від операцій на вищих рівнях ієрархії.

Фактично, функціонування моделі є обходом дерева моделі системи, що здійснюється за спеціальними правилами. Кожен крок обходу здійснюється під час обробки відповідної події. У листових вузлах дерева, що відповідають складним процесам, вузли замінюються на відповідні піддерева, і обхід продовжується на глибших рівнях ієрархії. У листових вузлах дерева, що відповідають конкретним активностям, відбувається виконання робочого навантаження моделі. У вузлах, що відповідають операціям алгебри процесів, відбувається планування подій для дочірніх вузлів у відповідності до семантичних правил операцій. Такий спосіб виконання моделі відрізняється від алгоритмічного, послідовного виконання операцій моделювання, який прийнятий у переважній більшості мов моделювання й скоріше відповідає функціонуванню предикативних мов логічного програмування, в яких машина логічного доведення здійснює обходження дерев логічних виразів для досягнення мети доведення. При цьому порядок виконання елементів моделі залежить не від порядку їхнього опису при задаванні моделі, а скоріше від правил обходу виразів та їхнього змісту.

Отже, головною перевагою задавання моделі у вигляді виразів алгебри процесів є компактність її опису та можливість строгого доведення взаємної подібності з об'єктом моделювання. Для всебічної оцінки переваг даного підходу розглянемо порівняння описаного програмного комплексу з відомими засобами моделювання у відповідності до певного набору критеріїв.

Порівняння середовища ПМАП із відомими засобами моделювання

Серед загальноприйнятих способів підвищення рівня технології створення моделей можна виділити наступні.

1. Створення таких формальних засобів опису об'єкта моделювання, які забезпечують взаємно однозначний зв'язок між компонентами системи та моделі.
2. Задавання параметрів, що визначають структуру компонентів імітаційної моделі.

3. Задавання компонентів моделі з високою алгоритмічною складністю на до-
вольній алгоритмічній мові (блочне моделювання).

4. Використання спеціальних конструкторів, що дозволяють побудову моделі
за допомогою ієрархічного підключення підмоделей.

5. Перегляд проміжних результатів функціонування моделі та статистичних
параметрів.

6. Використання реального робочого навантаження.

7. Можливість реалізації на паралельних і розподілених обчислювальних сис-
темах.

Використавши згадані способи підвищення рівня технології створення моде-
лей як критерії для порівняння, порівняємо запропонований засіб імітаційного
моделювання з відомими. Результати порівняння зведені в таблицю.

Порівняння з відомими засобами моделювання							
Назва засобу моделювання	Критерії						
	1	2	3	4	5	6	7
PIPE	-	-	-	+/-	+	-	-
PRISM	-	-	-	+	-	-	-
PEPA Workbench	+	-	-	+/-	-	-	-
HYDRA	+	+/-	-	+	-	-	-
UPPAAL	+	+	+	+/-	+	-	-
ПМАП	+	+	+	+	+	+	+

Комплекс PIPE — система імітаційного моделювання, яка реалізована на базі
узагальнених стохастичних мереж Петрі [6]. Тому модель у PIPE задають у вигляді
відповідного дводольного графа, що складається з вершин і переходів. Система
дозволяє виконувати модель у покроковому й пакетному режимах. Система також
забезпечує завантаження фрагментів мереж із бібліотеки, однак через обмеже-
ність застосованого формального засобу інтеграція цих фрагментів в існуючу мо-
дель є достатньо складною. У системі не передбачено задавання структурних па-
раметрів моделі й роботу з реальним робочим навантаженням.

Система PRISM — це засіб моделювання, в основу якого покладено оригіна-
льний формальний опис моделей, що не має безпосереднього зв'язку із предст-
авленням системи у вигляді маркованої системи з переходами [7]. Тому еквівалент-
ність об'єктів моделі та компонентів системи не може бути строго обґрунтована.
Усі критерії, окрім підтримки ієрархічного подання моделі, не виконуються.

PEPA Workbench базується на стохастичній алгебрі процесів PEPA [8]. Оскі-
льки для цього формального засобу доведені теореми еквівалентності, то відпові-
дність моделі може бути строго обґрунтована. У даній системі моделювання існує
підтримка ієрархічності, оскільки для алгебри процесів характерна підтримка цьо-
го критерію. Решта критеріїв не реалізовані.

HYDRA — це система моделювання, що базується на PEPA [9]. Відмінностя-
ми її від PEPA Workbench є повноцінна підтримка ієрархічності опису моделей, а
також можливість формування структури моделі на основі заданих параметрів. Це
є єдиним покращенням із точки зору розглянутих критеріїв.

UPPAAL — інтегрований комплекс імітаційного моделювання, основою якого є система таймованих автоматів [10]. Формальний засіб, покладений в основу системи, забезпечує еквівалентність об'єкта моделювання й моделі. Внутрішня мова задавання моделі дозволяє конструювати структуру моделі на основі параметрів. Особливості реалізації забезпечують можливість використання зовнішніх модулів для опису структури моделі. Також у комплексі обмежено реалізована підтримка ієрархічності опису моделей. Система імітаційного моделювання дозволяє покрокове й пакетне виконання моделі. Єдиним недоліком даного комплексу є відсутність підтримки реального робочого навантаження.

В якості прототипу для порівняння виберемо UPPAAL, як найближчий за критеріями. Проведемо порівняння складності представлення моделі в середовищах UPPAAL і ПМАП. Складність будемо визначати як кількість об'єктів N_{obj} , необхідних для реалізації моделі однієї й тієї ж системи. Результати порівняння наведені на рис. 2, де через N позначено кількість компонент системи, що підлягають моделюванню.

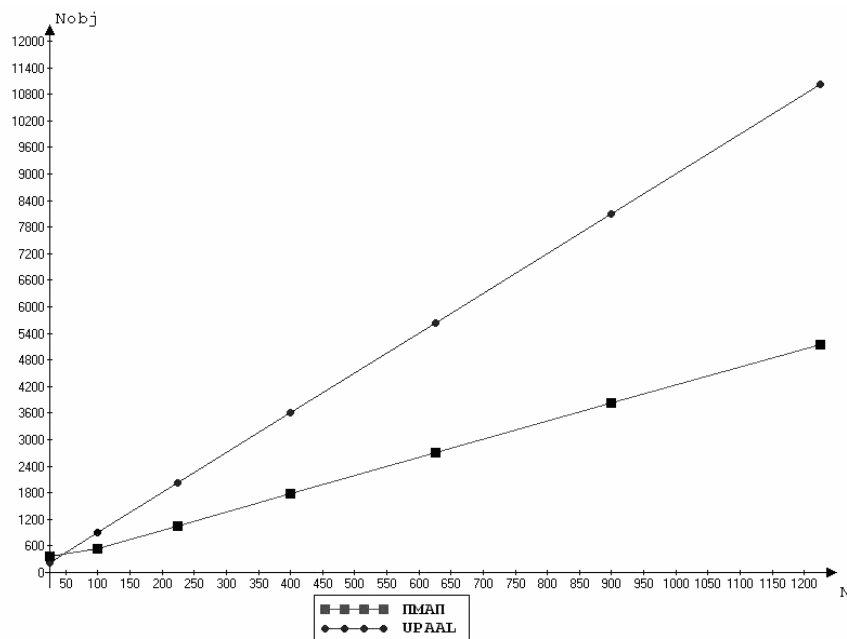


Рис. 2. Залежність кількості об'єктів моделі від розміру системи, що моделюється

Рис. 2 демонструє перевагу запропонованого підходу, в основі якого лежить повноцінно ієрархічний формальний засіб із динамічним способом задавання активних об'єктів моделі. Через це в ході моделювання не виникає необхідності зберігати в оперативній пам'яті всі об'єкти моделі, а лише активні об'єкти паралельних гілок моделі. UPPAAL, що оснований на автоматному формальному засобі зі статичною структурою моделі, повинен зберігати всі об'єкти моделі в процесі моделювання.

Проведемо ще одне порівняння представлених систем моделювання. Для цього заміряємо час одного прогону імітаційної моделі на кожному з комплексів моделювання. Побудуємо залежність часу одного прогону від кількості компонент системи, що моделюється. Одержані залежності зображені на рис. 3.

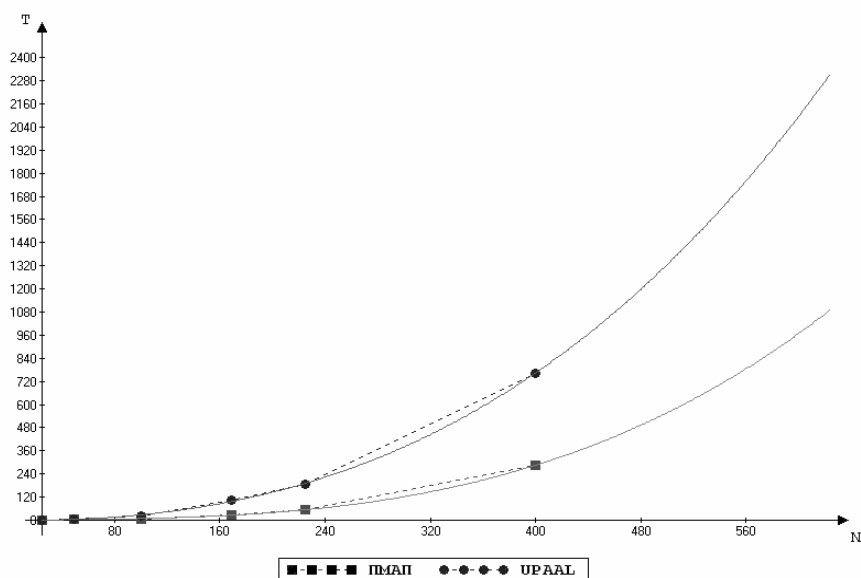


Рис. 3. Залежність часу одного прогону моделі від розміру системи, що моделюється

Рис. 3 демонструє часові переваги запропонованого комплексу перед UPPAAL. Форма залежностей співпадає, оскільки в обох випадках використовувався прогон моделі однієї й тієї ж системи з однією й тією ж крайовою задачею в якості робочого навантаження. Але переваги ПМАП щодо кількості об'єктів моделі дали можливість скоротити час для їхньої обробки.

Ще одна перевага програмного комплексу ПМАП полягає в тому, що він забезпечує можливість реалізації моделей на паралельних і розподілених обчислювальних системах.

Висновки

При побудові програмних комплексів для моделювання складних систем одним з актуальних завдань є вибір формального засобу опису об'єктів дослідження. Запропонований програмний комплекс базується на спеціально створеній версії алгебри процесів, яка представлена алфавітом і сукупністю операцій. Кожній операції алгебри процесів відповідає операція препроцесорної мови алгебри процесів. Базовий набір операцій включає префіксацію, паралельну композицію із синхронною взаємодією, паралельну композицію з асинхронною взаємодією, вибір та операцію приховування.

Мова ПМАП реалізована за допомогою програмного комплексу, який включає підсистему лексичного й синтаксичного розбору та підсистему імітаційного моделювання. Лексичний аналіз забезпечує формування виразів алгебри процесів на основі текстів мови ПМАП, а синтаксичний аналіз формує дерево розбору, що задає зв'язки між процесами та активностями. Структура підсистеми імітаційного моделювання включає ядро, чергу подій, чергу заблокованих процесів, масив складних процесів, масив змінних і масив міток. Функціонування моделі є обходом дерева моделі системи, що здійснюється за спеціальними правилами. Кожен

крок обходу здійснюється під час обробки відповідної події. Такий спосіб виконання моделі відрізняється від послідовного виконання операцій, який прийнятий у переважній більшості мов моделювання, оскільки порядок виконання елементів моделі залежить не від порядку їхнього опису при задаванні моделі, а скоріше від правил обходу виразів і їхнього змісту. Застосування сукупності згаданих підходів дозволило створити програмний комплекс, що має кращі характеристики в порівнянні з відомими середовищами моделювання на заданому наборі критеріїв.

1. *Баранов С.И.* Синтез микропрограмных автоматов. — Л.: Энергия, 1979. — 232 с.
2. *Бусленко Н.П.* Моделирование сложных систем. — М.: Наука, 1978. — 400 с.
3. *Нестеренко Б.Б., Новотарский М.А.* Мультипроцессорные системы. — К.: Институт математики АН Украины, 1995. — 408 с.
4. *Valiante G.* Algorithms on Trees and Graphs. — Berlin: Springer, 2002. — 450 p.
5. *Нестеренко Б.Б., Новотарский М.А.* Алгебра процессов для моделирования параллельных асинхронных вычислительных структур // Электронное моделирование. — 2006. — Т. 28, № 4. — С. 47–64.
6. *Bonet P., Llado C.M., Puijaner R., Knottenbelt W.J.* PIPE v2.5: A Petri Net Tool for Performance Modelling // Proc. 23rd Latin American Conference on Informatics, 2007. — 12 p.
7. *Kwiatkowska M.Z., Norman G., Parker D.* Probabilistic Symbolic Model Checking with PRISM: a Hybrid Approach // Software Tools for Technology Transfer. — 2004. — Vol. 6, N 2. — P. 128–142.
8. *Hillston J.* Compositional Approach to Performance Modeling. — Cambridge: Cambridge University Press, 1996. — 186 p.
9. *Bradley J.T., Dingle N.J., Gilmore S.T., Knottenbelt W.J.* Extracting Passage Times from PEPA Models with the HYDRA Tool: A Case Study // Proc. 19th UK Performance Engineering Workshop, 2003. — P. 79–90.
10. *Larsen K.G., Pettersson P., Yi W.* UPPAAL in a Nutshell // Int. Journal on Software Tools for Technology Transfer. — 1997. — Vol. 1, N 1–2. — P. 134–152.

Надійшла до редакції 17.10.2007