

РОЗПОДІЛЕНЕ МОДЕЛЮВАННЯ В EMS НА ОСНОВІ АРХІТЕКТУРИ HLA

Анотація. Пропонується технологія інтеграції потужного формального апарата E-мереж та ієрархічного агрегатного підходу, застосованого в системі імітаційного моделювання EMS, в архітектуру високого рівня HLA для проведення розподіленого моделювання з метою дослідження систем зі складною структурою.

Ключові слова: розподілене моделювання, система EMS, E-мережі, ієрархічний агрегатний підхід, архітектура HLA.

Аннотация. Предлагается технология интеграции мощного формального аппарата E-сетей и иерархического агрегатного подхода, примененного в системе имитационного моделирования EMS, в архитектуру высокого уровня HLA для проведения распределенного моделирования с целью исследования систем со сложной структурой.

Ключевые слова: распределенное моделирование, система EMS, E-сети, иерархический агрегатный подход, архитектура HLA.

Abstract. The technology of integration of the powerful formal E-nets device and hierarchic aggregate approach which is used in the system of simulation use of EMS, in High Level Architecture (HLA) for distributed modeling applied by research of complex structure systems is proposed.

Keywords: distributed modeling, EMS system, E-nets, hierarchic aggregate approach, HLA architecture.

1. Вступ

Роль моделювання як методу наукових досліджень та методу рішення технічних задач завжди оцінювалась досить високо. Особливого значення моделювання набуває в сучасних умовах прискорення науково-технічного прогресу – досить часто постає потреба проведення дослідження складних систем за умов обмеженості часових, трудових, енергетичних та матеріальних ресурсів.

Сучасні тенденції в галузі інформаційних технологій ставлять на порядок денний питання створення розподілених систем імітаційного моделювання, здатних на новій технологічній основі реалізовувати додаткові переваги моделювання як методу дослідження складних систем.

Застосування розподіленого моделювання дозволить:

- зменшити час реалізації імітаційних експериментів;
- збільшити об'єм пам'яті для рішення задачі;
- об'єднати різномірні системи імітаційного моделювання, моделі для яких розроблені та протестовані, в єдину розподілену середу імітаційного моделювання;
- використовувати географічно розподілені компоненти;
- інтегрувати програми моделювання, реалізовані різними розробниками;
- зменшити вплив збоїв у роботі комп'ютерів на процес моделювання в цілому;
- здавати в «оренду» не використані в даний момент часу обчислювальні ресурси;
- вести одночасну розробку декільком дослідникам однієї моделі.

Перспективною на сьогоднішній день є технологія розподіленого моделювання високого рівня High Level Architecture (HLA) [1]. Вона була розроблена Міністерством оборони США в кінці 90-х років з метою забезпечення взаємодії всіх типів моделей і підтримки їх багаторазового використання і визначається стандартами IEEE 1516, 1516.1, 1516.2. «Високий рівень» у назві означає, що компонентами моделювання є не об'єкти або логічні процеси, як в послідовному та розподіленому імітаційному моделюванні, а самі імітаційні моделі. HLA являє собою сукупність методик та стандартів для побудови розподілених

систем, включаючи забезпечення взаємодії учасників моделювання різних типів. Однак, поряд з даними перевагами, HLA орієнтована на використання об'єктних мов високого рівня, що ускладнює можливість її застосування, зокрема, для роботи з мережевими імітаційними моделями. Крім того, в HLA залишаються відкритими питання детального відпрацювання формальних методів і практичних рішень, що приймаються під час реалізації розподілених моделей.

Виходячи із важливості викладеної проблеми, нами була розроблена система розподіленого імітаційного моделювання E-net Modeling System (EMS) [2]. Для створення моделей у системі EMS застосовується потужний формальний апарат E-мереж [3] та ієрархічний агрегатний підхід. EMS має зручний web-інтерфейс, який забезпечує можливість географічно віддаленим користувачам сумісно одночасно працювати з однією імітаційною моделлю. E-мережі дозволяють відображувати не лише потоки управління, а й потоки даних, проводити їх кількісну обробку на переходах мережі, надають зручні механізми маршрутизації розвитку процесів, значно перевищують інші мережеві методи в реалізації логічних функцій. У свою чергу, використання ієрархічного агрегатного підходу дозволяє підійти до дослідження складних систем з позицій системного аналізу. Поряд з цим залишаються відкритими питання застосування даного формального апарата для проведення розподіленого моделювання на базі архітектури HLA.

Метою статті є викладення особливостей реалізації розподіленого моделювання в системі EMS, а також опис технології інтеграції E-мережових ієрархічних моделей в архітектуру високого рівня HLA.

2. Система імітаційного моделювання EMS

2.1. Архітектурні особливості системи

E-net Modeling System (EMS) являє собою web-систему імітаційного моделювання, основу архітектури якої (рис. 1) складають декілька серверів, які, взаємодіючи між собою, дозволяють створювати, редагувати, відлагоджувати моделі та проводити експерименти у розподіленому середовищі.

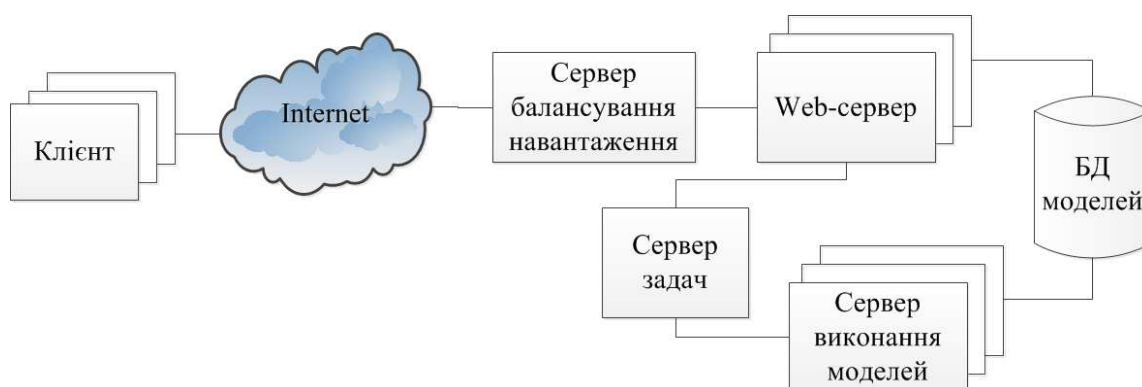


Рис. 1. Архітектура системи EMS

Завдяки наявності зручного web-інтерфейсу системи, розробленого на основі технології Vaadin [4], вдається спростити та прискорити розробку складних моделей географічно-віддаленим користувачам. Тобто, використовуючи лише браузер, користувач має можливість створювати, запускати на виконання та отримувати результати роботи моделей. Можливість забезпечення багатокористувацького режиму роботи з системою потребує наявності декількох web-серверів, навантаження між якими рівномірно розподіляється за допомогою серверу балансування навантаження. Створення імітаційної моделі, завантаження та збереження моделі, формування експерименту та опрацювання його ре-

зультатів забезпечуються підсистемою інтерфейсу, що виконується виключно на web-серверах. Імітаційні моделі потребують великої кількості обчислювальних ресурсів, що може призвести до перенавантаження web-серверу. Таким чином, для забезпечення ефективного використання ресурсів і надійної роботи web-серверів необхідно використовувати окремий сервер виконання моделей, в задачі якого не входить безпосередня обробка запитів користувачів. Підсистеми серверу виконання моделей показані на рис. 2.

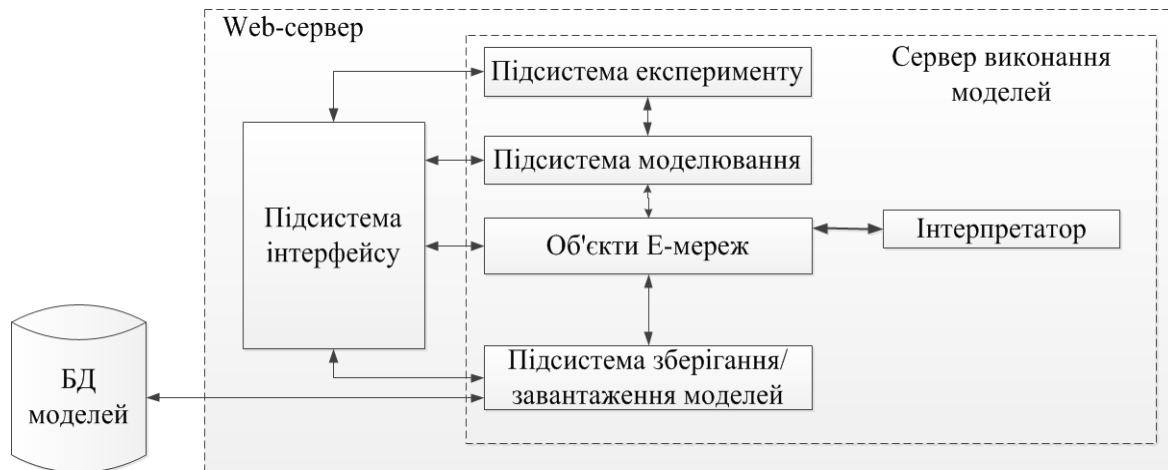


Рис. 2. Структура серверу виконання моделей

Сервер задач, у свою чергу, виконує рівномірний розподіл задач між окремими серверами виконання моделей. У БД моделей, що доступна всім web-серверам системи, зберігаються імітаційні моделі, створені користувачами, результати проведених імітаційних експериментів, а також дані про користувачів системи.

2.2. Формальний апарат

Оснoву формального представлення моделей в EMS складають Е-мережі, які є найбільш потужним розширенням мереж Петрі [3].

Е-мережа визначається п'ятіркою

$$E = \langle P, T, I, O, \mu \rangle,$$

де $P = \langle P_s, P_r \rangle$ – кінцева множина позицій мережі, що складається із множин P_s – простих позицій та P_r – вирішальних позицій; множини не перетинаються;

T – кінцева непуста множина переходів;

$I: P \times T \rightarrow \{0, 1\}$ – вхідна функція, що задає відображення позицій у переходи, тобто визначає для кожного $t \in T$ множину його вхідних позицій $I(t)$. Якщо існує дуга, що веде з $p \in P$ в $t \in T$, то $I(p, t) = 1$, в іншому випадку $I(p, t) = 0$;

$O: P \times T \rightarrow \{0, 1\}$ – вихідна функція, що задає відображення переходів мережі в позиції, тобто визначає для кожного $t \in T$ множину його вихідних позицій $O(t)$. Якщо існує дуга, що веде з $t \in T$ в $p \in P$, то $O(p, t) = 1$, в іншому випадку $O(p, t) = 0$;

$\mu: P_s \rightarrow \{0, 1\}$ – функція початкового маркування мережі.

Динамічні властивості Е-мережі визначаються зміною маркування мережі і залежать від значень компонентів управляючого відображення. Маркуванням управляючої Е-мережі будемо називати вектор $M = (M(p_1), M(p_2), \dots, M(p_n))$, де $n = |P_s|$. Позиція $p_i \in P_s$ називається вільною (не містить мітку), якщо $M(p_i) = 0$, в іншому випадку, при

$M(p_i) = 1$, позиція вважається зайнятою. Для заданого маркування M множину маркованих позицій будемо визначати як $P_M = \{p \in P_S \mid M(p) > 0\}$.

Основні особливості Е-мереж, що, фактично, визначають їх як клас обчислювальних мереж, полягають у розширених можливостях використання маркерів, позицій та переходів.

Кожній мітці, що знаходиться в позиції Е-мережі, ставиться у відповідність кортеж числових атрибутів, який визначає інформаційний зміст мітки $d_i = (d_{i1}, d_{i2}, \dots, d_{ij}, \dots, d_{iN})$, де d_{ij} – значення j -го атрибута i -й мітки. Під час переміщення міток по мережі значення їх атрибутів можуть змінюватись. При виконанні мережі мітки можуть переходити з вхідних позицій переходів у вихідні, змінюючи маркування мережі.

При моделюванні систем зі складною структурою також необхідно вирішити протиріччя між простотою опису взаємодії учасників моделювання та необхідністю врахування широкого спектра особливостей їх функціонування. Рішення даної проблеми пропонується в [5], завдяки спільному використанню Е-мереж та агрегатного підходу, в основі якого лежить поняття агрегатної системи, що складається з певним чином пов'язаних між собою агрегатів, виділених на етапі декомпозиції системи. З точки зору моделювання, агрегат виступає як достатньо універсальний оброблювач інформації, який може приймати вхідні та видавати вихідні сигнали. Функціонування системи в цілому складається з автономних процесів роботи окремих агрегатів, внутрішня структура яких описується за допомогою Е-мереж, та обміну сигналами між ними.

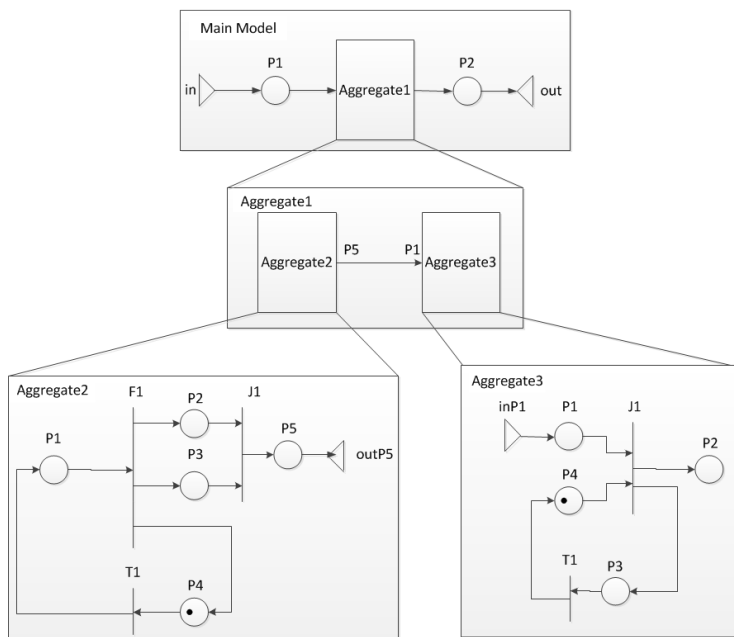


Рис. 3. Структура пірамідально зростаючої агрегатної моделі

наковою структурою, але з різними параметрами. Введення поняття типу агрегату як сукупності всіх агрегатів з однією структурою надає можливість зберігати опис структури один раз. Крім того, у разі необхідності, можлива миттєва зміна в різних моделях усіх агрегатів одного типу, якщо під час розробки моделі користувачем були внесені корективи в його структуру. Для створення моделей в EMS розроблено спеціальний графічний редактор, що дає можливість розроблювати легко сприймані моделі та позбавляє від необхідності вивчення спеціальної мови програмування, чим значно спрощує процес моделювання систем зі складною структурою. Розроблені моделі зберігаються в форматі XML-файлів, з підтримкою стандарту PNML [6], який був нами розширений з урахуванням таких особливостей використання Е-мереж:

- наявність переходів та черг різних типів;

Зазначимо, що в системі EMS запропоновано апарат пірамідально зростаючих агрегатних моделей. Тобто, з однієї сторони, до складу кожного агрегату може входити декілька інших агрегатів, що дозволяє створювати моделі шляхом їх поступового ускладнення. На рис. 3 зображена структура пірамідально зростаючої агрегатної моделі.

Також в EMS введено поняття типу агрегату, що продиктовано можливістю використання в різних моделях агрегатів з однаковою структурою, але з різними параметрами.

Введення поняття типу агрегату як сукупності всіх агрегатів з однією структурою надає можливість зберігати опис структури один раз.

- можливість визначення функцій на кожному переході;
- наявність різних типів атрибутів міток;
- наявність пірамідально зростаючих агрегатів.

У розширеному стандарті PNML кожному типу агрегату відповідає один файл з безпосереднім описом структури та XML-файл спеціально розробленого формату з описом відмінностей (початкове маркування, значення функцій на переходах, параметри) між різними агрегатами одного типу. В свою чергу, використання стандарту PNML для зберігання моделей на основі Е-мереж дозволяє EMS підтримувати сумісність з іншими системами імітаційного моделювання, що використовують формальний апарат мереж Петрі.

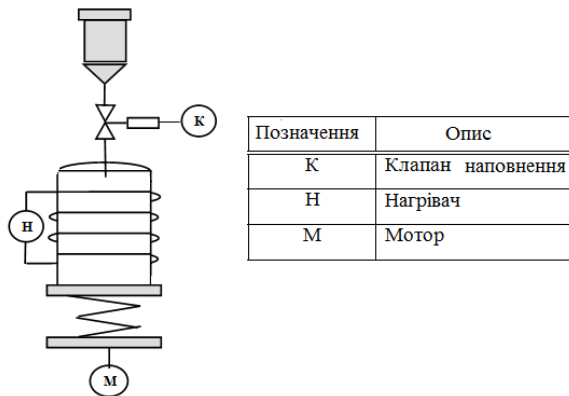


Рис. 4. Схема процесу наповнення інерційної капсули

2.3. Створення моделей

Детально розглянемо створення моделей у системі EMS на прикладі моделювання процесу заповнення інерційної капсули, який описаний у [7]. Особливість даного процесу полягає в тому, що заповнення капсули порошком повинно проходити поступово з одночасним струшуванням та нагрівом капсули. При цьому температура капсули під час наповнення повинна підтримуватись у заздалегідь визначених умовах. Структурна схема даного процесу наведена на рис. 4.

Дамо короткий опис модельованого процесу. У вихідному стані капсула пуста, клапан наповнення закрито. Після встановлення та фіксації капсули на вібраторі вмикається нагрівач. При досягненні нижньої границі температури вмикається вібратор, потім відкривається клапан наповнення. Якщо під час наповнення капсули порошком температура досягне верхньої границі, нагрівач відключається і вмикається знову при досягненні нижньої границі температури. В той час, як капсула заповниться порошком, клапан закривається і не раніше, ніж через 1 хвилину, вимикається нагрівач і зупиняється мотор вібратора.

Представимо модель даного процесу, розроблену в системі EMS, у вигляді трьох взаємодіючих агрегатів (рис. 5).

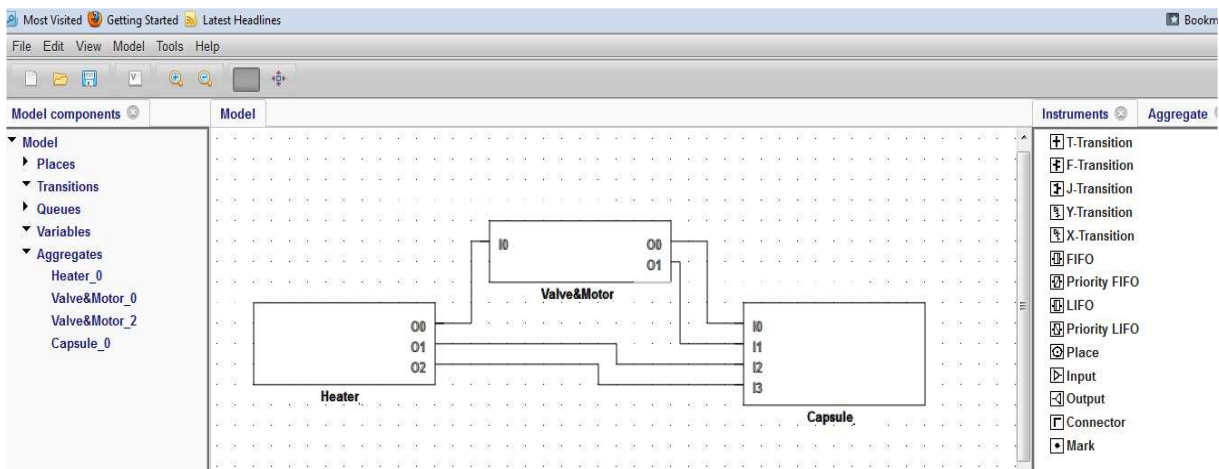


Рис. 5. Агрегатна модель процесу наповнення капсули

Два агрегати представляють елементи управління: перший агрегат (HEATER) забезпечує управління нагріванням, другий (VALVE&MOTOR) – клапаном наповнення та

мотором вібратора. Третій агрегат (CAPSULE) моделює капсулу, що нагрівається та запо-внюється порошком. Використовуючи графічний редактор системи EMS, спроектуємо фо-рмалізовані Е-мережіві схеми агрегатів (рис. 6–8).

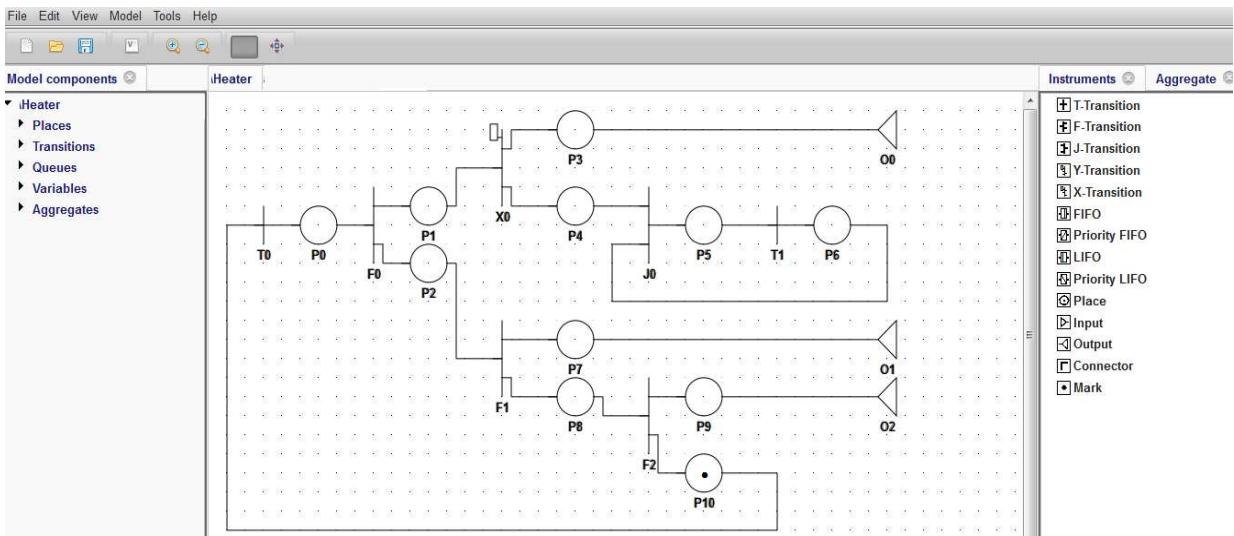


Рис. 6. Е-мережова модель управління нагрівачем

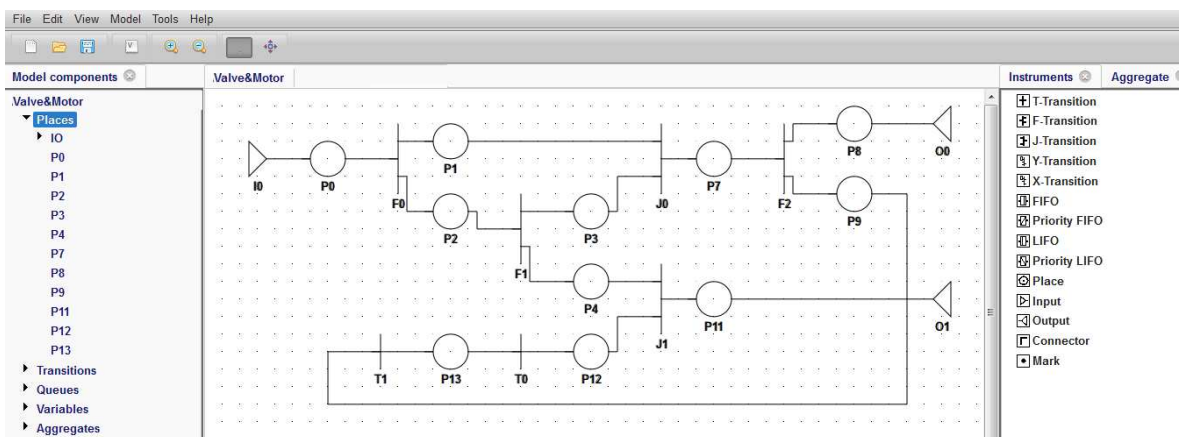


Рис. 7. Е-мережова модель управління клапаном наповнення та мотором

У початковому стані жодна з позицій даного агрегату не маркується. Мітка посту-пає в позицію P0 після включення нагрівача.

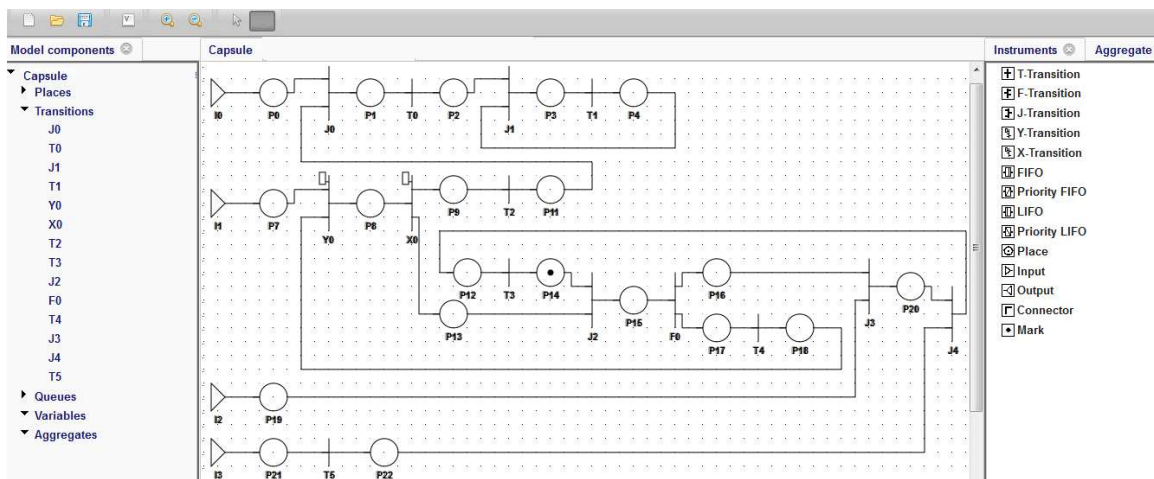


Рис. 8. Е-мережова модель капсули

Даний агрегат імітує стан капсули під час роботи алгоритмів управління. В початковому стані маркірована позиція P14 – агрегат готовий до роботи з нагрівачем. Мітки поступають від агрегату «HEATER» в позицію P7 (після включення нагрівача) та в позицію P0 (після відключення нагрівача), а також від агрегату «VALVE&MOTOR» в позиції P19 (після відкриття клапану наповнення) та в позицію P21 (після відключення нагрівача, мотора та закриття клапана).

3. High Level Architecture: концепція та основні переваги використання

Технологія розподіленого моделювання високого рівня HLA поєднує системи, побудовані для різних цілей у різні періоди часу та з різною структурою, продукти та платформи різних фірм, дозволяє їм взаємодіяти в єдиному синтетичному оточенні. Можливість включення окремих елементів, тренажерів, керованих персоналом, цілих систем обладнання (верстатів, машин, контролерів, плат тощо) в цикл моделювання не тільки позбавляє від необхідності імітації роботи складних технічних пристроїв та поведінки людей, але й знижує ймовірні похибки в моделюванні фізичних процесів, що в значній мірі підвищує надійність одержуваних результатів.

Ключовими поняттями в HLA є федерація (Federation) та RTI (Run-Time Infrastructure) [8]. Федерація – це сукупність компонентів імітаційного моделювання, які називаються федератами (federates), що співпрацюють для вирішення деякої конкретної задачі. В різних експериментах одні й ті самі федерати можуть входити до складу різних федерацій. Технологія HLA не накладає обмежень на внутрішню структуру федератів, але визначає стандарт для опису інформації про об'єкти імітаційного моделювання – шаблон об'єктних моделей (Object Model Template – ОМТ). Даний стандарт забезпечує взаємодію федератів та дає можливість їх багаторазового використання незалежно від внутрішньої структури в різних федераціях для проведення моделювання. Таким чином, усі об'єкти федерації повинні бути описані на основі ОМТ. RTI забезпечує взаємодію федерацій і федератів, містить у собі загальнодоступні сервіси для координування взаємодії учасників моделювання, а також підтримує різні способи синхронізації всередині федерації.

В HLA інформація, якою обмінюються федерати, може бути двох типів:

- 1) атрибут об'єкта;
- 2) інтеракція.

Кожен об'єкт у будь-який момент часу характеризується своїм станом, що визначається набором поточних значень його атрибутів. Федерат, управляючий даним об'єктом (атрибутом об'єкта), може змінити його стан, змінивши значення атрибута. За допомогою сервісів RTI федерат передає нове значення атрибута іншим федератам – федерат оновлює атрибут; федерат, що приймає нове значення, відображає атрибут.

Інтеракції, на відміну від стану об'єктів, не підтримуються в системі постійно, вони мають миттєву природу і являють собою дію, що виконується об'єктом федерата та спричиняє можливі зміни стану об'єкта іншого федерата. Наприклад, постріл (миттєва дія) із зброя (об'єкт, що виконує дію) по цілі (об'єкт, на стан якого може вплинути дія).

Як було зазначено вище, весь обмін даними між федератами відбувається через RTI. Механізм обміну реалізовано у вигляді «підписки», тобто федерат, зацікавлений в отриманні певних атрибутів, повинен підписатися на них за допомогою сервісів RTI. Слід зазначити, що з метою зниження завантаження каналів RTI відслідковує тільки змінення даних. Як результат, інтеракції передаються завжди, а атрибути тільки при зміні їх значень.

Таким чином, можна сказати, що в моделюванні HLA відіграє ту саму роль, що і такі відомі технології, як CORBA, COM+, DCOM та ін. Детальне порівняння HLA з іншими технологіями розподіленого моделювання проведено в [9], на основі якого можна зробити висновок, що використання розподіленої архітектури HLA надає такі переваги:

- об'єднання різноманітних систем імітаційного моделювання, моделі для яких вже розроблені та протестовані;
- інтеграція програм моделювання, реалізованих різними розробниками;
- проведення одночасної сумісної розробки географічно віддаленим користувачам з однією імітаційною моделлю;
- вибір різноманітних схем управління часом у процесі моделювання, зокрема, режиму реального часу;
- використання географічно розподілених компонент однієї моделі;
- використання різних даних для однієї моделі в залежності від мети моделювання (завдяки відокремленню в HLA даних від архітектури);
- вибірковий розподіл інформації між взаємодіючими моделями, що, у свою чергу, дає можливість для ефективного масштабування систем;
- модифікація механізмів розподіленого моделювання для забезпечення їх відповідності конкретним задачам (завдяки використанню відкритих стандартів);
- взаємодія всіх типів моделей завдяки універсальності об'єктної ідентифікації та об'єктних зв'язків.

4. Інтеграція формального апарата EMS в архітектуру HLA

Кожна імітаційна модель EMS розглядається як один з федератів, що, у свою чергу, відноситься до федерації – сукупності моделей системи, яка досліджується.

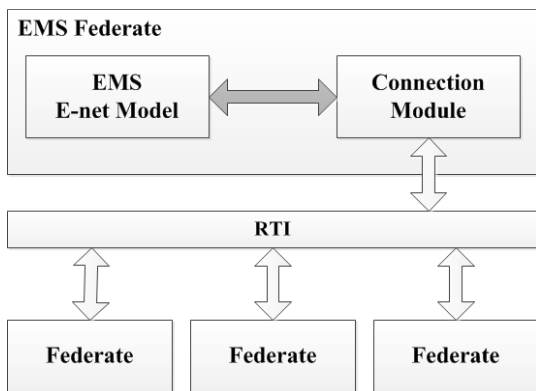


Рис. 9. Взаємодія федератів EMS в архітектурі HLA

Для забезпечення взаємодії імітаційних моделей, створених у системі EMS, у рамках архітектури HLA були розроблені спеціальні модулі – Connect Module (CM), що виконують єдину функцію між моделями та RTI під час моделювання. CM приймає атрибути та інтеракції, які надходять від інших федератів, передає атрибути федератам згідно з механізмом підписки та синхронізує локальний час виконання кожного федерата згідно з глобальним часом роботи всієї федерації. Таким чином, модель, розроблена в системі EMS, взаємодіє з RTI HLA через модуль CM (рис. 9).

Невід'ємною складовою процесу моделювання є збір статистичних даних. Під час проведення розподіленого моделювання задача збору статистичних даних для кожної імітаційної моделі покладається на підсистему експерименту моделі. Зібрана статистика також передається через модуль CM.

Взаємодія з сервісами RTI забезпечується наявністю в CM унікальних входів та виходів (RTIInput, RTIOutput відповідно) і на основі використання методів основних інтерфейсів RTI: RTIAmbassador та FederateAmbassador. При потраплянні мітки на вхід RTIInput спрацьовує метод settoken, що встановлює в атрибутах мітки E-мережі новий атрибут, прийнятий від іншого федерата, або оновлює атрибут. Також при необхідності можуть бути змінені значення входних змінних (параметрів) мережі. В CM реалізована можливість формування черг (списків) повідомлень. Дана особливість продиктована необхідністю синхронізації роботи федератів. Так як кожен федерат працює в своєму локальному часі, який, звичайно, не співпадає з локальним часом іншого федерата, виникає необхідність формування списків повідомлень та зберігання часу їх отримання й необхідного виконання, щоб не втратити повідомлення з часовою міткою, більшою за модельний час федерата та не пропустити ті, в яких часова мітка менша. Таким чином, під час передачі повідомлення, крім значень атрибутів та змінних, передається часова (як атрибут мітки) міт-

ка, що вказує час виконання оновлення атрибутів і синхронізує роботу федератів. Відповідно, при потраплянні мітки на вихід RTIOutput метод settoken перевизначається для передачі повідомлень у RTI.

5. Реалізація інтерфейсів HLA

Для забезпечення виконання розроблених у системі EMS моделей у розподіленому середовищі необхідно реалізувати методи інтерфейсів, специфікації яких надають стандарти HLA, та використати доступні сервіси RTI.

Для RTI нами було обрано програмну реалізацію PORTICO, що є відкритою та надає сервіси для управління федерацією, деклараціями, об'єктами, правом доступу, розподіленням даних та часом. Особливість використання PORTICO полягає в необхідності використання конфігураційного файлу RTI.RID (Run-Time Infrastructure Declaration), в якому знаходяться всі основні налаштування, що використовує RTI під час запуску. Зазначимо, що даний файл повинен знаходитись у директорії, де виконується федерат. Альтернативою даного рішення є встановлення системної змінної RTI_RID_FILE.

Взаємодія та обмін даними між федератами відбувається за допомогою сервісів RTI з використанням двох основних інтерфейсів:

1. RTI Ambassador – забезпечує виклик функції під час запиту сервісу RTI федератом. Даний інтерфейс знаходиться в бібліотеці LibRTI.

2. Federate Ambassador – надає необхідні функції (зворотного виклику – callback functions), що застосовує RTI під час відправлення повідомлень та відповідей до федерата. Всі callback functions реалізовані в федераті як підклас класу FederateAmbassador, що знаходиться в бібліотеці LibRTI.

Розглянемо реалізацію методів головних інтерфейсів HLA в системі EMS на прикладі роботи одного федерата, життєвий цикл якого складається з таких етапів:

1. Створення Federate Ambassador та RTI Ambassador для федерата, під час якого необхідно вказати як параметри назву федерації та файл, що містить об'єктну модель федератів:

```
File fom = new File ("ModelFed.xml");
rtiAmb.createFederationExecution("ModuleFederation");
fom.toURI().toURL();
```

2. Включення в федерацію. В функцію передається назва федерата, назва федерації, в яку включають федерат та об'єкт класу Ambassador даного федерата:

```
rtiAmb.joinFederationExecution(fedName, "CapsuleFederation", fedAmb).
```

3. Синхронізація роботи федерата з роботою федерації за допомогою точки синхронізації:

```
rtiAmb.registerFederationSynchronizationPoint(syncPoint, null);
while( fedAmb.isAnnounced == false )
{
    rtiAmb.tick();
}
rtiAmb.synchronizationPointAchieved(syncPoint).
```

Як параметр передається назва точки синхронізації (syncPoint), яку створює перший федерат у новоствореній федерації.

4. Очікування виконання синхронізації інших федератів даної федерації:

```
log("Achieved sync point: "+syncPoint+ ", waiting for federation...");
while( fedAmb.isReadyToRun == false )
{
    rtiAmb.tick();
}
```

У разі успішного завершення синхронізації значення глобальної змінної isReadyToRun стане true під час відповіді функції зворотного виклику.

5. Публікація та підписка класів і атрибутів:

```
private static void publishAndSubscribe() throws RTIException{
int classHandle = rtiAmb.getObjectClassHandle( "ObjectRoot.Modell" );
int paHandle = rtiAmb.getAttributeHandle( "atr1", classHandle );
AttributeHandleSet attributes = RtiFactoryFactory.getRtiFactory().
createAttributeHandleSet();
attributes.add( paHandle );
//can use with time
rtiAmb.publishObjectClass( classHandle, attributes );
rtiAmb.subscribeObjectClassAttributes(classHandle, attributes);}.
```

6. Реєстрація екземплярів класів та їх атрибутів:

```
private int registerObject() throws RTIException
{
    int objectHandle=rtiAmb.getObjectClassHandle("objectRoot.Capsule");
    return rtiAmb.registerObjectInstance(objectHandle).
}
```

7. Виконання роботи моделі (федерата):

```
ModelExample example = new ModelExample();
example.main(null).
```

8. Знищення зареєстрованих об'єктів:

```
rtiAmb.deleteObjectInstance(objectHandle, generateTag()).
```

9. Вихід з федерації:

```
rtiAmb.resignFederationExecution(ResignAction.NO_ACTION).
```

На рис. 10 зображена загальна архітектура федерації розглянутої вище моделі інерційної капсули. Кожний агрегат представлено як федерат, що, взаємодіючи з іншими федератами розподіленої моделі, утворює федерацію. Крім розглянутої вище структури агрегатів, представленої у вигляді E-мережі, кожен федерат містить у собі СМ, за допомогою якого відбувається взаємодія моделей з RTI.

Файл ModelFed.xml містить шаблон об'єктних моделей федерації та федератів. Даний файл використовується RTI при створенні федерації. В ньому визначена вся структура обміну даними між федератами.

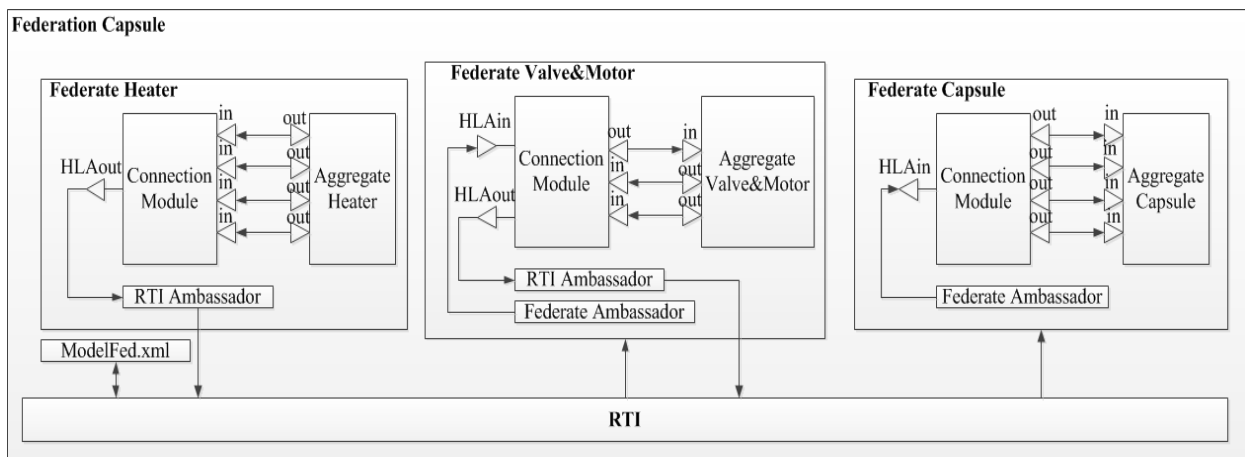


Рис. 10. Загальна структура федерації

Дана модель може бути використана для оцінки параметрів, які дозволять детально дослідити систему за різних умов роботи. Результати проведення експериментів відображаються в формі графіків, таблиць та гістограм. На рис. 11 наведені результати моделювання, представлені у вигляді таблиці, які, поряд з іншими даними, відображають середню кількість спрацювання клапана (перехід T3), мотора моделі (перехід T5), а також, скільки разів вмикався нагрівач під час проведення моделювання (перехід J2).

The screenshot shows a window titled 'Report' with a table of transition statistics. The table has four columns: 'Name of transition', 'Occupied coefficient', 'Number of passed tokens', and 'Average occupied time'. The data is as follows:

Name of transition	Occupied coefficient	Number of passed tokens	Average occupied time
J0	0.0	1.0	0.0
J1	0.0	0.0	0.0
J2	0.0	9.0	365.0
T3	0.0	6.0	201.0
T4	0.0	0.0	0.0
T5	0.0	3.0	319.0

Рис. 11. Статистичні дані проведення експерименту

6. Висновки

Наведена технологія інтеграції потужного формального апарата Е-мереж та ієрархічного агрегатного підходу з архітектурою HLA дозволяє застосувати розподілену схему використання системи імітаційного моделювання EMS з усіма притаманними цій схемі перевагами. Додаткові модулі, що реалізують інтерфейси федерацій, забезпечують синхронізацію моделей та збір статистичних даних стандартними засобами RTI, а графічний інтерфейс та його XML-інтерпретація додають необхідної універсальності процесу створення й дослідження моделей.

Описана технологія та інструментальні засоби можуть бути в подальшому використані для побудови синтетичного оточення інформаційних систем з метою виявлення й уточнення їх властивостей та характеристик, у тому числі і за допомогою методу напівнатурного моделювання.

СПИСОК ЛІТЕРАТУРИ

1. Fujimoto R.M. Distributed Simulation Systems / Fujimoto R.M. – NY: A Wiley-Interscience publication, 2000. – 303 p.
2. Казимир В.В. Розподілена система імітаційного моделювання EMS / В.В. Казимир, Г.А. Сіра, І.І. Мушкетик // Вісник Чернігівського державного технологічного університету. – 2011. – № 3. – С. 144 – 153.
3. Nutt G. Evaluation Nets for Computer Systems Performance Analysis / G. Nutt // FJCC, AFIPS PRESS. – 1972. – P. 279 – 286.
4. Gronroos M. Book of Vaadin: 4th Edition / Gronroos M. – NY: VaadinLtd, 2011. – 434 p.
5. Казимир В.В. Моделирование синтетического окружения для реактивных систем / В.В. Казимир // Математичне моделювання. – 2003. – № 2 (10). – С. 24 – 32.
6. The Petri Net Markup Language: Concepts, Technology and Tools [Електронний ресурс] / J. Billington, S. Christensen, Kees van Hee [et al.]. – Режим доступу: <http://xml.coverpages.org/PNML200303.pdf>.
7. Казимир В.В. Модельно-ориентированное управление интеллектуальными производственными системами: дис. ... доктора техн. наук: 05.13.06 / Казимир Владимир Викторович. – К., 2006. – 301 с.
8. Замятина Е.Б. Современные теории имитационного моделирования: Программа специального курса / Замятина Е.Б. – Пермь: ПГУ, 2007. – 119 с.
9. Современные технологии построения распределенных программных систем / А.П. Афанасьев, А.И. Ваньков, А.И. Волошинов [и др.]. – М.: Проблемы системного анализа и управления, 2001. – 53 с.

Стаття надійшла до редакції 19.10.2011