

УДК 004.9:504:519.6

МЕТОДИКА ІНТЕГРАЦІЇ МАТЕМАТИЧНИХ МОДЕЛЕЙ НАВКОЛИШНЬОГО СЕРЕДОВИЩА ДО СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Є.О. Євдін

Інститут проблем математичних машин і систем НАН України

e-mail: yewgen@env.com.ua

ВСТУП

Для розв'язання задач екологічної безпеки та раціонального використання природних ресурсів все ширше впроваджуються системи підтримки прийняття рішень (СППР), засновані на математичних моделях динаміки навколишнього природного середовища. Зазвичай математичні моделі розробляються різними науковими колективами, на різних мовах програмування поза зв'язку з розробкою СППР, без урахування їх сумісності з іншими моделями. Тому актуальною є задача незалежної інтеграції моделей до системи, при якій логіка моделі ізольована від логіки системи та зміни в моделі мало або взагалі не впливають на логіку ядра СППР. Теоретичні засади розв'язання такої задачі були розроблені в роботі 1 на підставі запропонованого у названій роботі «об'єкту-обгортки». Метою даної роботи є розробка практичних рекомендацій щодо інтеграції моделей у СППР.

РОЗПОДІЛЕНИЙ ОБ'ЄКТ-ОБГОРТКА

Розподілений об'єкт-обгортка моделі (РОО) [1] – це програмний об'єкт комунікації СППР і моделі, що розподіляється під час виконання між різними компонентами системи і забезпечує логічну, візуальну та технічну інтеграцію математичної моделі в СППР. Структура РОО представлена на рис. 1. Основним і обов'язковим компонентом обгортки моделі є ModelWrapper. Це єдиний компонент, який оперує як даними у форматі системи, так і даними у форматі моделі. Другим компонентом РОО є ModelEntryPoint – компонент, відповідальний за технічну частину роботи з інтеграції зовнішньої моделі. Він реалізує виклик методів моделі. Спосіб виклику, в першу чергу, залежить від реалізації моделі у вигляді програмного об'єкта (автономна програма, бібліотека, що динамічно підключається, веб-сервіс, програма, завантажена на кластер тощо). У загальному випадку введення вхідних даних обчислювальних моделей здійснюється через графічний інтерфейс користувача, що реалізується компонентом ModelUserInterface.

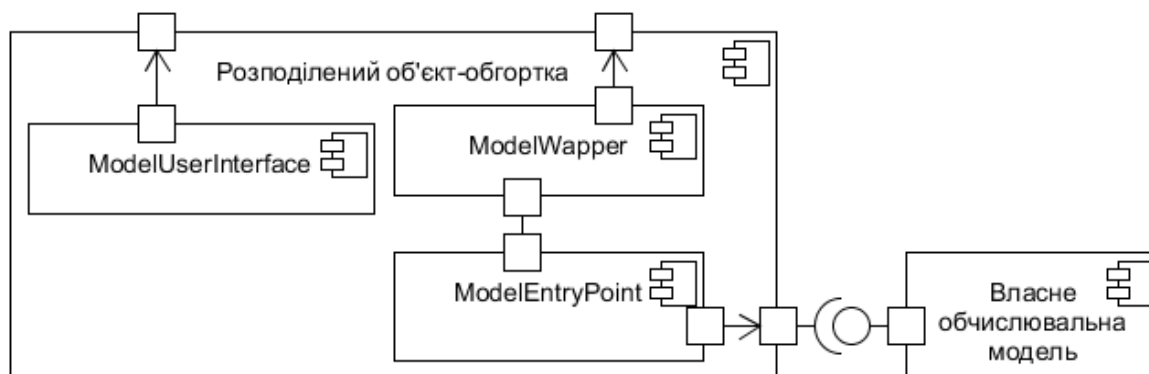


Рис. 1. Структура РОО моделі

РОО дозволяє ізолювати логіку моделі від системи. З одного боку, РОО слідує всім вимогам по стандарту та інтерфейсу, які пред'являються СППР до інтегрованих додатків, а з іншого – управляє таким додатком з використанням відповідних технологій.

МЕТОДИКА ІНТЕГРАЦІЇ

Процедуру інтеграції нової моделі в готову систему підтримки прийняття рішень можна розглядати, як окремий проект з розробки програмного забезпечення. Дійсно, мають бути визначені завдання та цілі інтеграції, бажана функціональність, створений програмний продукт (обгортка моделі, модифікований код розрахункової моделі), протестована якість інтеграції, і модель у вигляді плагіна має бути впроваджена в систему. При цьому процедура інтеграції має свої особливості, на підставі яких можна запропонувати модель життєвого циклу (ЖЦ) інтеграції математичних моделей. Ця модель ЖЦ адаптована до умов завдання і повинна допомогти розробникам, що інтегрують модель (інтегратори моделі), впоратися із завдання в короткі терміни.

Модель ЖЦ інтеграції є варіантом моделі ітераційної розробки [2], який полягає в розділенні процесу на послідовність етапів або ітерацій, на кожному з яких створюється працездатний прототип, що послідовно наближається за функціональністю до необхідного продукту. Ітераційний процес дозволяє локалізувати і виправляти помилки інтеграції послідовно, в міру поглиблення інтеграції системи і моделі. Основну роботу, яку необхідно провести на кожному етапі, можна формулювати заздалегідь і описані нижче етапи інтеграції мають деяку надлишкову функціональність, пов'язану з підтримкою зумовлених наборів вхідних даних. Проте саме така черговість кроків дозволяє послідовно збільшувати глибину інтеграції, отримуючи на кожному кроці готовий до використання системою прототип моделі, що дозволяє легко виявляти помилки інтеграції характерні для кожного з етапів.

Вихідною точкою для початку процесу інтеграції моделей є наявність працездатної, верифікованої версії коду розрахункової моделі (наприклад, у вигляді автономного програмного продукту, що читає вхідні файли і формує вихідні файли). Додатково, інтегратор моделі потребує документацію з описом структури вхідних та вихідних даних моделі, а також регламентом її функціонування в СППР. Наявність результатів початкової версії розрахункової моделі дозволить тестувати правильність інтеграції моделі шляхом порівняння результатів інтегрованої моделі та початкової програми. Схематичне зображення роботи моделі представлено на рис 2. Тут і надалі при описі ЖЦ світлим хрестиком позначені вхідні дані і їх передача, темним хрестиком – вихідні дані та їх передача.

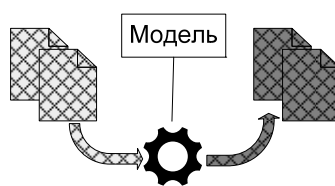


Рис. 2. Початковий стан інтеграції моделі

На першому етапі (рис. 3) проводиться інтеграція моделі на низькому рівні, а саме технічна організація взаємодії система – модель. Тобто, відбувається створення обгортки моделі (реалізуються два компоненти ModelWrapper і ModelEntryPoint), визначення способу технічної інтеграції розрахункового коду і обгортки моделі, модифікація (при необхідності) коду розрахункової моделі для забезпечення цієї інтеграції та реалізація зв'язку система – модель для підтвердження правильності вибраного рішення. Рекомендується реалізувати функціональність запуску розрахунку моделі на основі наперед визначених вхідних даних із формуванням вихідних даних в початковому вигляді. Це дозволить протестувати правильність інтеграції шляхом порівняння результатами початкової версії моделі.

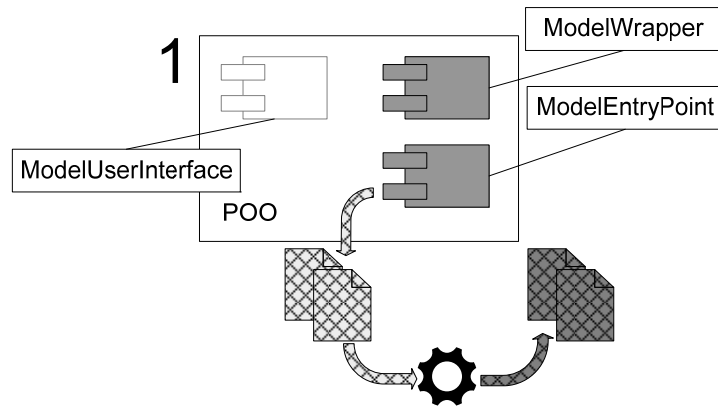


Рис. 3. Перший етап інтеграції моделі

На другому етапі (рис. 4) в систему інтегруються вихідні дані моделі. Визначаються вихідні дані, їх структура, типи і форми представлення користувачеві, реалізується перетворення вихідних даних моделі в типи даних системи і формується дерево результатів. Як і на першому етапі, модель запускається на основі визначених вхідних даних, проте результат вже відображається засобами системи. Це надасть додаткову інформацію, окрім початкового виведення, для аналізу і верифікації результатів розрахунку розробникові моделі, що особливо важливо, якщо модель все ще знаходиться в розробці.

На третьому етапі (рис. 4) відбувається інтеграція вхідних даних моделі. Визначаються всі необхідні вхідні дані, їх типи і структура. Окремим завданням стоїть визначення джерел вхідних даних, будь то інтерфейс користувача, дані з автоматичних джерел (бази даних, ГІС та ін.) або з попередніх моделей. Формується дерево вхідних даних і реалізується перетворення цих даних у формат моделі. Додатково реалізується логіка управління розрахунком моделі залежно від вхідних даних і проміжних результатів. При цьому вхідні дані все ще можуть формуватися з набору наперед визначених даних, проте вже в компоненті ModelWrapper записуватися у формат уніфікованого типу даних, який пізніше прочитуватиметься в компоненті ModelEntryPoint. Таким чином, метою третього етапу є отримання закінченої функціональності компонента ModelEntryPoint.

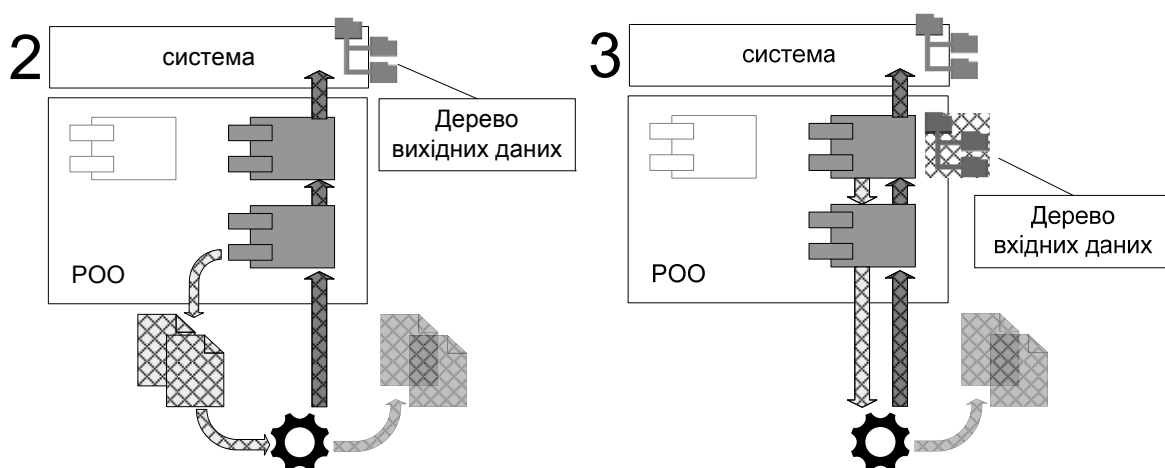


Рис. 4. Другий та третій етап інтеграції моделі

На четвертому етапі (рис. 5) реалізується функціональність наповнення вхідних даних: створюється інтерфейс користувача, бази даних і запити до них, проводиться попередня обробка вхідних даних, якщо в цьому є необхідність. Результатом четвертого етапу повинен стати повнофункціональний прототип інтегрованої моделі, який можна

видавати на розширене тестування. Відмітимо, що четвертий етап сам по собі є досить трудомістким, при цьому його можна виконувати паралельно з етапами 1–3. Спільним результатом етапів 1-3 та 4 є лише дерево вхідних даних.

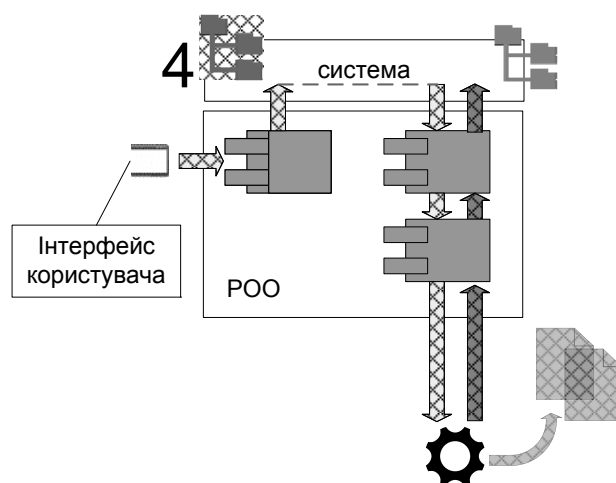


Рис. 5. Четвертий етап інтеграції моделі

На п'ятому етапі за результатами тестування прототипу вносяться зміни в обгортку моделі в будь-якій з її частин (додавання нового результату, нових вхідних параметрів, пост-обробка даних і так далі), що є фактично проведенням етапів 2–4 з новими параметрами. П'ятий етап повторюється до тих пір, поки не буде прийнято рішення про впровадження інтегрованої моделі в систему. Після завершення етапу, формування вихідних даних в початковому вигляді можна прибирати.

ВИСНОВКИ

В роботі запропоновано практичні рекомендації інтеграції моделей в СППР на підставі розподіленого об'єкту-обгортки та практичного досвіду інтеграції більше ніж 25 різних математичних моделей у Європейську СППР з реагування на ядерні аварії JRODOS [3]. Запропонована методика інтеграції нових моделей з використанням РОО дозволяє зменшити можливість появи помилок інтеграції, своєчасно їх локалізувати та виправити. Апробацією даної методики можна вважати те, що повний редизайн системи RODOS, розпочатий за бажанням користувачів даної системи [4], було здійснено за 3 роки, тоді як на момент початку редизайну розробка системи тривала більше 15 років.

Список літератури

1. Евдин Е.А. Технология интеграции математических моделей в системы поддержки принятия решений в сфере экологической безопасности на основе распределенных объектов-оберток / Е.А. Евдин // Электронное моделирование. – 2014. – 36, № 6. – С. 23-42.
2. Литвинов В.В., Голуб С.В., Григор'єв К.М., Жигульська В.Ю. Об'єктно-орієнтовне моделювання при проектуванні вбудованих систем і систем реального часу. Навч. посібник «Системний аналіз та проектування комп'ютерних інформаційних систем». — Черкаси: Вид. від. ЧНУ ім. Б. Хмельницького, 2011 – 376 с.
3. RODOS reengineering: aims and implementation details / Ie. Ievdin, D. Trybushnyi, M. Zheleznyak, W. Raskob // Radioprotection. – 2010. – Vol. 45, N5. – P. 181 – 189.
4. Gering F., Raskob W., 2007. Experience with the use of RODOS in emergency centers and reasons for the re-design of RODOS. 36. Пр. науково-практичної конф. з міжнар. участю СППР 2007, Київ, 7 червня 2007 р. – С. 26-28.