

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ РОЗПОДІЛЕНОГО МОДЕЛЮЮЧОГО СЕРЕДОВИЩА ДЛЯ ДОСЛІДЖЕННЯ СКЛАДНИХ ДИНАМІЧНИХ ПРОЦЕСІВ

Анопрієнко О.Я., Абабнех Х., Забровський С.В., Малихін В.О.

Кафедра ЕОМ ДонНТУ

anoprien@cs.dgtu.donetsk.ua

Abstract

Anoprienko A., Ababneh H., Zabrovsky S., Malykhin V. Realization features of distributed simulation environment for research of complex dynamic processes. New universal architecture on the base of Java-technology for distributed simulation environment is proposed and described.

Вступ

В період с 1997 року по 2005 рік в ДонНТУ виконувалась серія розробок та досліджень з метою удосконалення розподілених моделюючих середовищ для дослідження складних динамічних процесів [1-7]. Одним з основних напрямків при цьому був розвиток інтерактивних можливостей системи DIVA [8] та забезпечення її ефективного використання у сучасному мережному середовищі.

При цьому, серед іншого, були вирішені наступні питання:

- автоматизація створення файлів з ініціалізацією значень параметрів моделювання (*start, *.anl, *.dat, *.do) та зміни деяких параметрів при моделюванні хімічного процесу;
- можливість аналізу процесу, за допомогою завдання діапазону зміни якого-небудь параметра і подальшого перегляду стану системи в діапазоні зміни параметра;
- можливість наочного керування процесом моделювання.

На першому етапі був розроблений відносно спрощений графічний інтерфейс і впроваджені розподілені обчислення. Але перший варіант розподіленого моделюючого середовища страждав такими недоліками як платформозалежність клієнтської частини розподіленого додатка, орієнтованість інтерфейсу користувача на одну фіксовану модель, неможливість масштабування, необхідність постійного спостереження за відновленнями інтерфейсу користувача системи DIVA.

Ряд недоліків, властивий першому етапові, був виправлений у другому варіанті розподіленого моделюючого середовища. А саме,

клієнтська частина стала платформонезалежною, завдяки використанню технології JavaApplet відпала необхідність у спостереженні за відновленням версії клієнтського прикладання, набагато зменшився обсяг трафіка при моделюванні. Однак, недоліки все-таки залишилися. Інтерфейс розподіленого моделюючого середовища так і залишився залежним від моделі процесу.

Тому на наступному етапі з метою підвищення ефективності та універсальності архітектури моделюючої системи були запропоновані нові вимоги до розподіленого середовища моделювання:

- незалежність клієнтського програмного забезпечення від платформи;
- розподіл обчислень з покладанням всієї обчислювальної частини на сервер;
- незалежність від процесу, що моделюється;
- легкість у взаємодії із серверною частиною при написанні клієнтської;
- територіально розподілене середовище;
- відсутність необхідності спостереження користувачем за відновленням середовища розроблювачами.

Удосконалене розподілене моделююче середовище для системи моделювання DIVA

Організація розподіленого моделюючого середовища для системи моделювання DIVA у відповідності з клієнт-серверною архітектурою передбачає певні правила та команди для взаємодії між компонентами середовища, сукупність яких і складає протокол взаємодії клієнтської та серверної частин.

Використання клієнт-серверної архітектури передбачає наявність серверної частини, розташованої на сервері і безпосередньо взаємодіючої з системою.

Загальна структура моделюючого середовища представлена на рисунку 1.

У загальному вигляді протокол побудовано на чергуванні команд від клієнтської частини програмного забезпечення та відкликів серверної частини з наступною передачею вихідних даних чи результатів моделювання відповідно від клієнтської частини до серверної та навпаки.

Серверна частина програмного забезпечення розподіленого середовища для системи моделювання DIVA виконує наступні команди:

- завантаження XML-файла з сервера на клієнт;
- запуск на виконання моделюючої системи DIVA;
- оновлення значень змінних для моделювання;
- аналіз та передача результатів моделювання клієнту;

- встановлення файлів загальної побудови моделі для системи DIVA у каталог моделювання;
- розрив з'єднання між клієнтом та сервером.

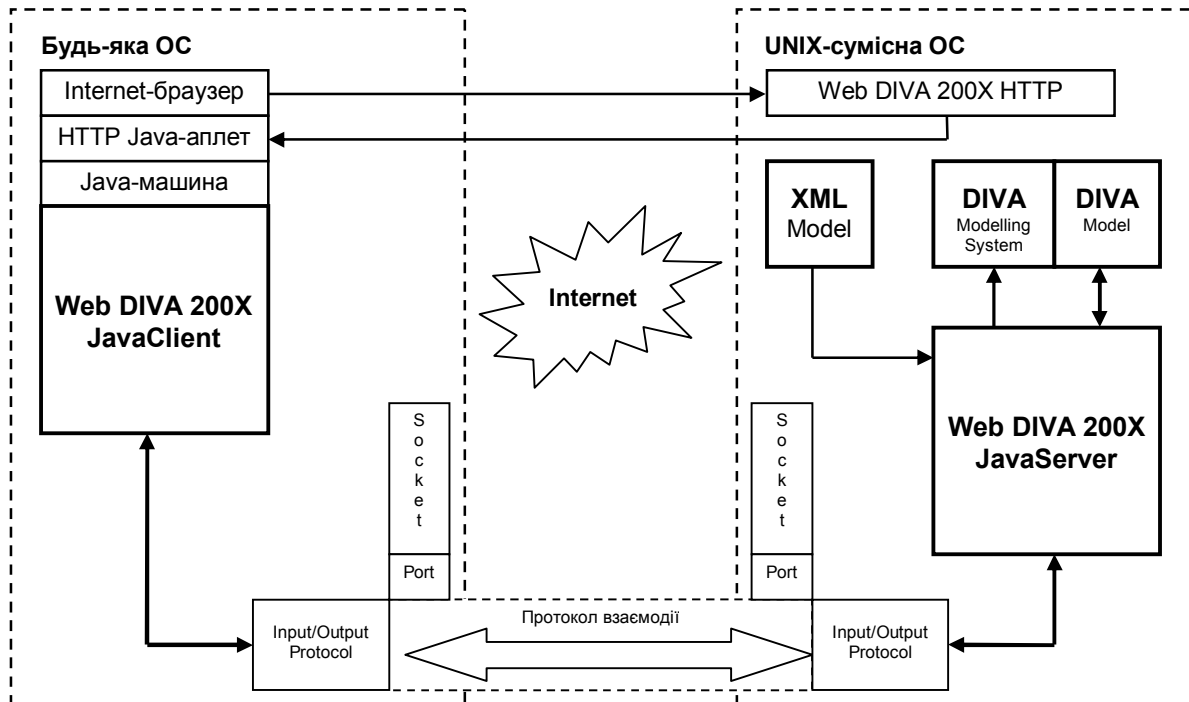


Рисунок 1 - Загальна структура моделюючого середовища

Команда завантаження XML-файла з сервера на клієнт призначена для отримання клієнтом XML-файла, який описує модель для системи моделювання DIVA.

Механізм ініціалізації моделюючого середовища побудований таким чином, що під час ініціалізації аплета клієнт викликає метод GetXML() DIVA-сервера, який дозволяє доступ до файлу опису моделі. Файл має формат xml і складається з набору списків, кожен з яких визначає відповідний блок моделі, де кожному елементу ставиться у відповідь набір полів що визначають діапазон зміни параметра, його опис та інша службова інформація. Наприклад, розглянемо фрагмент XML-файла:

```
<parameter name="Расход метанола (102)" min="5000" max="14000"
ed="kg/h">
  <description>Расход метанола в EA1101</description>
  <action>
    <target>f102.xliqpar</target>
    <mean>this/3600/32.04</mean>
  </action>
  <display x="2" y="298"
  <startValue>10000</startValue>
  <redzone1>7000</redzone1>
  <redzone2>11000</redzone2>
```

</parameter>

У загальному вигляді протокол можна представити у вигляді чергування посилки команд і підтверджень. Серверна частина програмного забезпечення розподіленого середовища для системи моделювання DIVA виконує наступні команди, надані клієнтською частиною:

- завантаження файлу з сервера на клієнт – призначена для отримання клієнтом файлу XML;
- встановлення файлів загальної побудови моделі для системи DIVA у каталог моделювання – призначена для зменшення трафіку від клієнтської до серверної частини програмного забезпечення шляхом зберігання на сервері та встановлення за запитом від клієнта у каталог моделювання незмінних даних щодо моделі по виробництву оцтової кислоти для НПО “Азот”;
- запуск на виконання моделюючої системи DIVA – запускає процес моделювання для файлів побудови моделі, передає моделюючій системі DIVA команди та отримує від неї і зберігає журнал виконання;
- аналіз, перетворення та передача результатів моделювання клієнту – виконує аналіз файлів результатів моделювання, перетворює присутні у них дані у формат об’єктів та передає отримані об’єкти до клієнтського програмного забезпечення;
- розрив з’єднання між клієнтом та сервером – призначена для розриву з’єднання після закінчення сеансу моделювання і відсилається як клієнтом, так і самим сервером у випадку помилкового розриву з’єднання.

Клієнт і сервер взаємодіють один з одним по встановленому протоколу. Розглянемо послідовність взаємодії:

1. Запуск сервера, відкриття відповідних сокета та порту, чекання приєднання клієнта
2. Запит від клієнта до сервера про установа з’єднання
3. Обмін вітанням
4. Чекання сервером надходження команд від клієнта
5. Упакування команди клієнтом у мережне повідомлення (об’єкт класа MessageWriter), відправлення повідомлення (об’єкт класа OutProtocol)
6. Прийом повідомлення з мережі сервером (об’єкт класа InProtocol), дешифрація отриманої команди
7. Відправлення сервером мережного повідомлення клієнту про вдале отримання команди
8. Виконання команди клієнта сервером
9. Відправлення сервером мережного повідомлення клієнту з результатом виконання отриманої команди

10. Відправлення клієнтом мережного повідомлення серверу про благополучне одержання результатів виконання команди
11. Виконання сеансу моделювання та аналізу результатів
12. Запит клієнта на завершення сеансу
13. Завершення сеансу сервером
14. Підтвердження сервером завершення сеансу

Клієнтська частина розрахована на виконання на базі віртуальної машини Java від Sun Microsystems.

Засоби графічної візуалізації разом з засобами, що реалізують протокол обміну з сервером необхідно оформити у вигляді Java-аплета із зручним для користувача інтерфейсом і зрозумілими принципами керування.

При розробці засобів візуалізації необхідно враховувати всі вимоги, а так само специфіку застосування розробленого програмного забезпечення, як складової частини розподіленого моделюючого середовища для моделювання складних хімічних процесів.

Вся розробка графічного інтерфейсу повинна бути направлена на те, щоб абстрагувати користувача від формату файлів даних, що описують модель в DIVA. Користувач спілкується тільки з візуальним інтерфейсом, а сама система моделювання DIVA і способи спілкування з нею реалізовувалися прозорими для користувача. Зрештою йому навіть не обов'язково знати про її існування і про платформу, на якій вона працює.

Особливості взаємодії моделюючого середовища з системою моделювання DIVA

У разі потреби виконання команди клієнта сервером моделюючого середовища, клієнт створює об'єкт Java-класу *OutProtocol* пакета *WebDIVA2004_Protocol*. Згідно до встановленого протоколу взаємодії клієнт запрошує виконання потрібної команди.

Об'єктом класа *OutProtocol* команда передається створеному об'єкту *MessageWriter*.

Об'єкт *MessageWriter* упаковує команду протокола до мережевого повідомлення. Мережеве повідомлення має формат UTF-рядка (UTF-string).

У розробленому моделюючому середовищі відправником повідомлення може розглядатися як клієнт, так і сервер.

Клієнт ініціює виконання команд сервером, тобто відправляє мережні повідомлення з командою серверу, а також опрацює повідомлення про успішне одержання результатів моделювання, відправлених йому сервером.

Сервер крім відправлення мережних повідомлень з результатами моделювання, також може відправляти повідомлення з підтвердженням одержання команди клієнта і повідомленням про виконання команди після відповідних дій.

Мережне повідомлення згідно з протоколом взаємодії моделюючого середовища має наступні поля (рисунок 2):

- “:” – відмітне поле ознаки (1 байт);
- *Size* – розмір тіла повідомлення чи даних команди (1 байт);
- *Type* – тип повідомлення (1 байт);
- *Data / Command* – поле тіла повідомлення чи даних команди (максимально 255 байт).

На прийомній стороні моделюючого середовища очікується надходження повідомлення з даними чи командою. З появою повідомлення зчитується з порту відкритого сокета на прийомній стороні і передається знов створеному об'єкту класу *MessageReader* для дешифрації команди, що надійшла, з боку відправника. Повідомлення розглядається з урахуванням зазначеного розміру тіла повідомлення з відповідного поля. З рядка береться тіло повідомлення. На наступному етапі відповідно до результату дешифрації повідомлення, створюється об'єкт класу *InProtocol* і, відповідно отриманій команді чи даним, викликається необхідний метод протоколу для виконання необхідних дій.

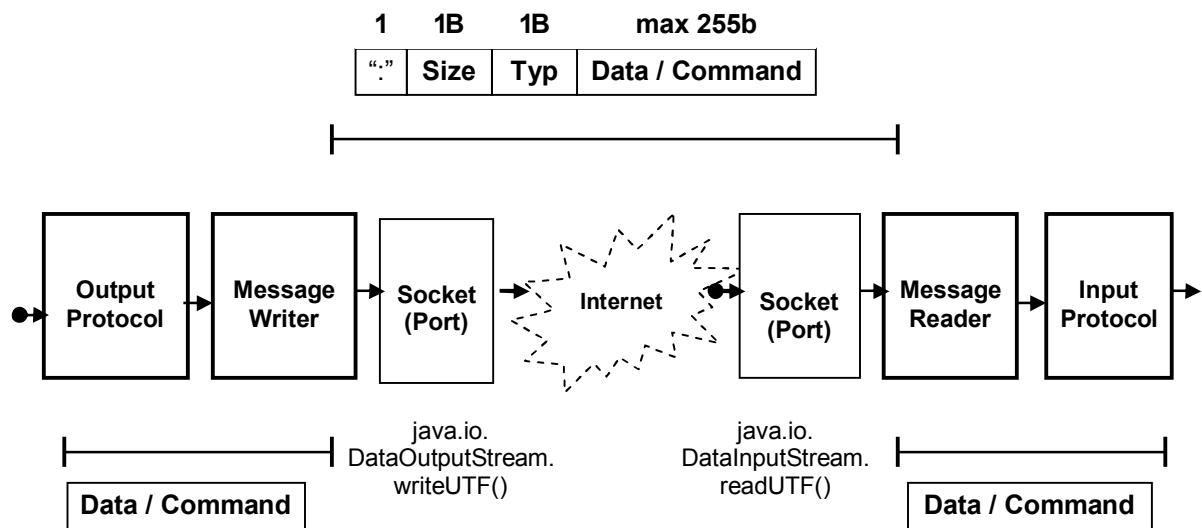


Рисунок 2 - Схема запропонованого протокола взаємодії клієнт-сервер моделюючого середовища

Сервер виконує наступні команди клієнта згідно з встановленим протоколом розробленого моделюючого середовища (рисунок 3):

- *GetConnection()* - Запит на початок сесії.
- *SetModel()* – Встановлення копії моделі у каталог моделювання.
- *GetXML()* - Запит на отримання XML-файла з описом моделі процесу. Опис моделі у вигляді універсального XML-файла забезпечує можливість зберігання файлів моделі DIVA тільки на боці сервера.
- *Restate(BlockName.VarName,NewValue)* - Запит на виконання сервером оновлення початкових значень вказаних змінних у файлі “.stat”.
- *SetTime(NewTime)* - Встановлення часу моделювання (файл моделі DIVA “.do”).
- *Integrate()* - Запит на виконання моделювання у системі DIVA (запуск DIVA з файлової системи UNIX-сумісної ОС).
- *GetVar(BlockName.VarName)* - Запит останнього значення змінної (передача результатів моделювання із файлів “.dsf” та “.ddf”).
- *GetVarScope(BlockName.VarName)* - Запит масиву змінної (передача результатів моделювання із файлів “.dsf” та “.ddf”).
- *SetParameter(ParamName,NewValue)* - Встановлення значень параметрів у файлі “.dat”.
- *RestateLast()* - Запит на виконання сервером оновлення початкових значень змінних у файлі “.stat” значеннями змінних наприкінці останнього сеансу моделювання (поетапний режим моделювання)
- *Disconnect()* - Підтвердження завершення сеансу.

Висновки

Таким чином, запропонована загальна методика модернізації систем моделювання складних технологічних процесів. Розроблено засоби реалізації розподіленого моделюючого середовища для системи моделювання DIVA. При цьому розроблена нова модель структури системи моделювання складного технологічного процесу, побудованої на базі розподіленого моделюючого середовища, модифіковано загальну методику побудови розподілених систем моделювання складних технологічних процесів.

Розроблено універсальні засоби взаємодії клієнтської та серверної частин моделюючого середовища для подальшого використання їх як для системи DIVA, так і, з мінімальними модифікаціями, в інших складних системах. Розроблено універсальні компоненти моделюючого середовища, які можуть бути використані при подальшому проектуванні системи.

Дані засоби дозволяють при написанні клієнтської частини абстрагуватися від структури та деталей побудови самої середовища моделювання (наприклад DIVA). Це досягається тим, що у моделюючому середовищі розроблено уніфікований протокол мережевої взаємодії, який дозволяє шляхом відправлення команд від клієнта до сервера, керувати процесом моделювання, а також тим, що усі функції по обміну даними з системою моделювання виконуються сервером, який надає клієнту уніфікований протокол передачі команд та даних.

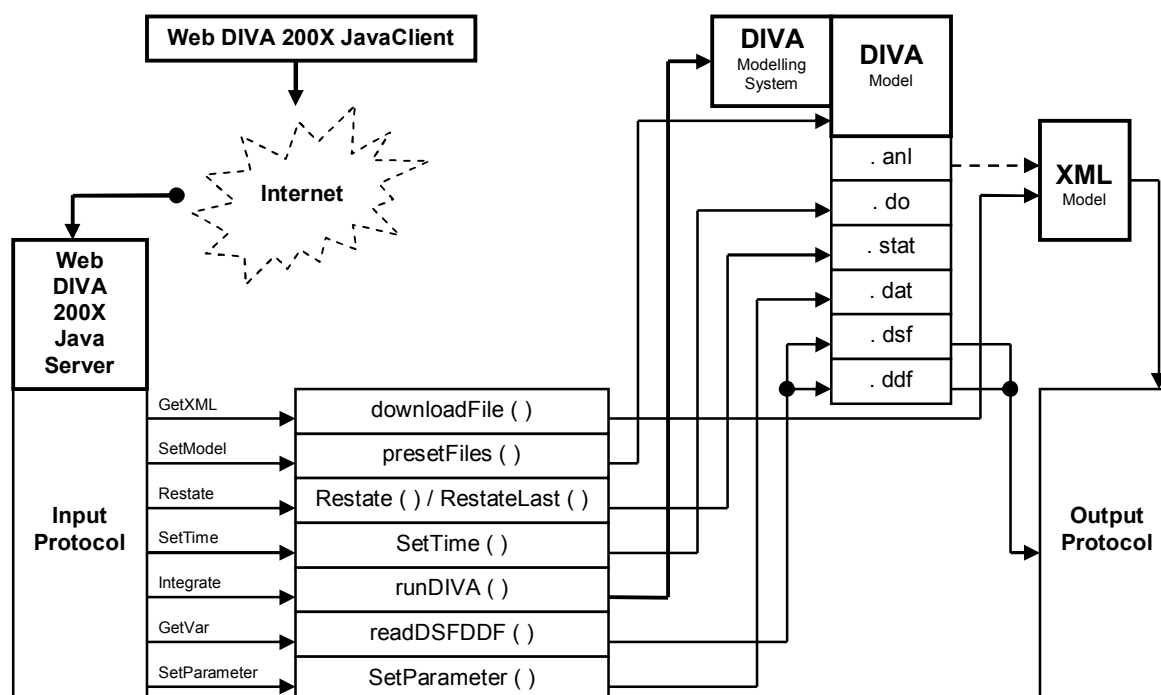


Рисунок 3 - Схема взаємодії JAVA-сервера з системою моделювання DIVA в ОС UNIX

Література

1. Аноприенко А. Я., Кинле А., Святный С. Н., Осипова Т. Ф. Моделювання реактора синтезу оцтової кислоти на базі моделюючої середовища DIVA // Збірник наукових праць Донгту. Серія "Інформатика, кібернетика й обчислювальна техніка". Випуск 1 (ИКВТ-97). - Донецьк: ДонГТУ. - 1997. - С. 16-21.
2. Забровський С.В. Методи і засоби графічної візуалізації в моделюванні складних систем. Випускна робота магістра. – Донецьк: ДонДТУ, 2000.
3. Каневский А.Д. Розробка засобів моделювання складних технічних процесів хімічного виробництва. Дипломний проект. – Донецьк: ДонДТУ, 2000.

4. Мацак С.А. Розробка та дослідження тренажерів складних технологічних процесів, орієнтованих на використання системи DIVA в мережному середовищі. Випускна робота магістра. – Донецьк: ДонТУ, 2001.
5. Сісюкін В.С. Методи та засоби реінженірінгу систем моделювання складних технологічних процесів на прикладі системи DIVA. Випускна робота магістра. – Донецьк: ДонТУ, 2001.
6. Шахов Сергій Михайлович. Розробка сучасних засобів розподіленого моделювання складних систем”. Випускна робота магістра. – Донецьк: ДонНТУ, 2002.
7. Шрамко Максим Васильович. Розробка графічних інтерфейсів для дослідження складних систем. Випускна робота магістра. – Донецьк: ДонНТУ, 2002.
8. Gilles E.D. DIVA. A Tool for Dynamic Simulation of Chemical Engineering Applications (<http://www.isr.uni-stuttgart.de/diva/diva.html>).

Дата надходження до редакції 21.09.2006 р.