

Лекція 17.

**Моделювання систем
з розподіленими параметрами**

Якщо об'єкт характеризується деяким параметром, різним за своїм значенням у різних крапках об'єкта, то можна сказати, що значення такого параметра розподілені (по об'єкті). Якщо таких параметрів трохи, то об'єкт розглядається як система з розподіленими параметрами. Для здійснення розрахунків систему в такому випадку зручно розбити на елементарні об'єми (шари). Покажемо це на прикладі.

Приклад 1. Розглянемо процес сушіння матеріалу (див. мал. 17.1). Допустимо, що матеріал являє собою купу сировини, над якою встановлений нагрівач. У процесі сушіння тепло нерівномірно проникає в глиб купи, і в різних шарах матеріал має різну температуру. Тобто в даному прикладі можна сказати, що параметр температури розподілений по об'єкті «купа». Опишемо кожен шар своїм рівнянням, використовуючи процедуру побудови моделі для окремого шару з лекції 11.

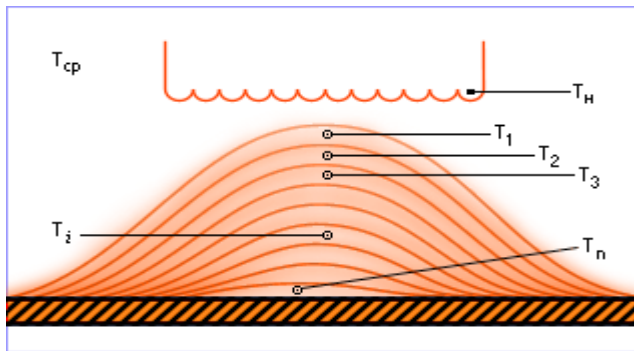


Рис. 17.1. Схема технологічного процесу сушіння матеріалу, розбитого на шари

Зміна температури в першому шарі (T_H — температура нагрівача, T_{cp} — температура середовища):

$$\frac{dT_1}{dt} = k_1 \cdot (T_H - T_1) + k_2 \cdot (T_2 - T_1) + k_3 \cdot (T_{cp} - T_1) \quad (1)$$

Зміна температури в i -ом шарі:

$$\frac{dT_i}{dt} = k_2 \cdot (T_{i+1} - T_i) + k_2 \cdot (T_{i-1} - T_i) \quad (2)$$

Зміна температури в n -ом шарі:

$$\frac{dT_n}{dt} = k_2 \cdot (T_{n-1} - T_n) \quad (3)$$

Тоді поведження системи «купи сировини» опишеться **системою диференціальних рівнянь**, кожне з яких опише окремий шар «купи».

Алгоритм розрахунку такої системи показаний на мал. 17.2. Особливістю алгоритму є те, що він містить на додаток до циклу за часом (див. алгоритм на мал. 10.5) вкладений цикл по номері шаруючи. Дійсно, на кожному такті необхідно окремо прорахувати зміни в кожному із шарів «купи». **Розподіленість об'єкта (введення додаткової координати) імітується додатковим циклічним блоком.**

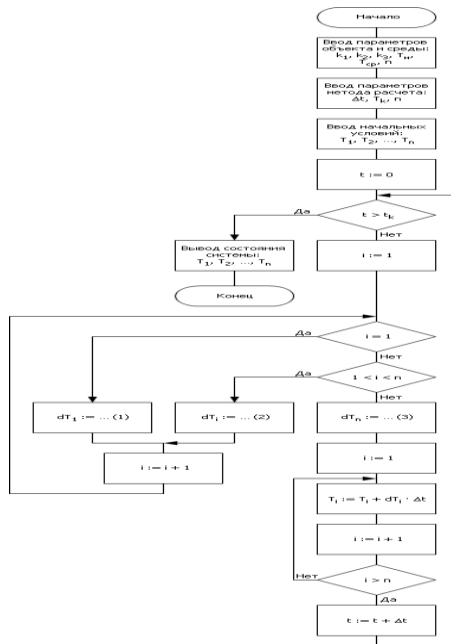


Рис. 17.2. Блок-схема моделювання розподіленої системи (на прикладі процесу сушіння)

Знову варто звернути увагу на те, що цикл розрахунку похідних і цикл розрахунку нового стану системи відділені друг від друга щоб уникнути появи «ефекту гонок».

Отже, *інструментом моделювання розподілених систем є вкладені цикли*, принаймні, подвійні — усередині циклу «за часом» утримується цикл «по просторі».

Модель може виглядати по-різному — залежно від платформи, на якій вона реалізується. Вище (див. мал. 17.2) був розібраний приклад, що показує алгоритмічний спосіб опису розрахунку поведження розподіленої системи, що використовує мову й архітектуру фон нейманівської машини (див. мал. 17.3).

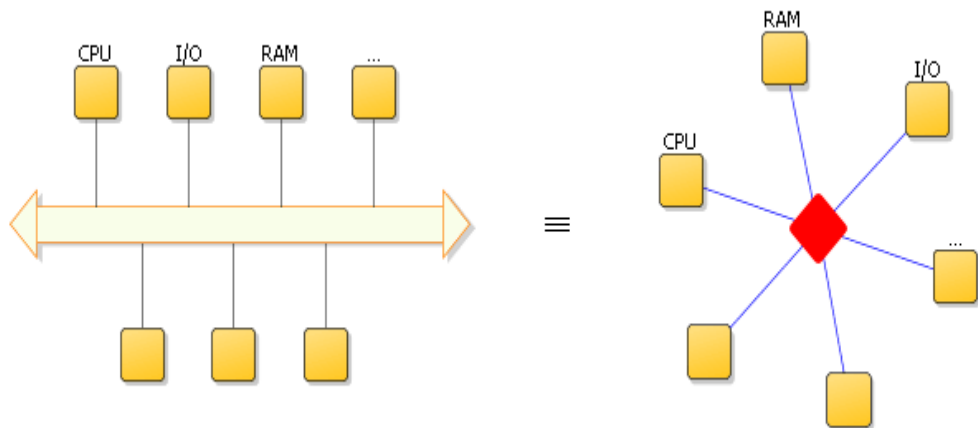


Рис. 17.3. Фон найманівська архітектура обчислювальної системи
(із загальною шиною, зірка)

Представте, що ми використаємо для моделювання структуру не фон нейманівської машини, що складається із **МНОЖИНИ** елементарних однакових обчислювальних елементів, з'єднаних між собою в мережу (див. мал. 17.4). При такій архітектурі елементи існують одночасно, тобто паралельно синхронно в часі й наприкінці кожного такту обмінюються вихідними (вхідними) сигналами між собою, сусідами по мережі. Така парадигма обчислювальної платформи імітує більш адекватно навколишній нас мир.

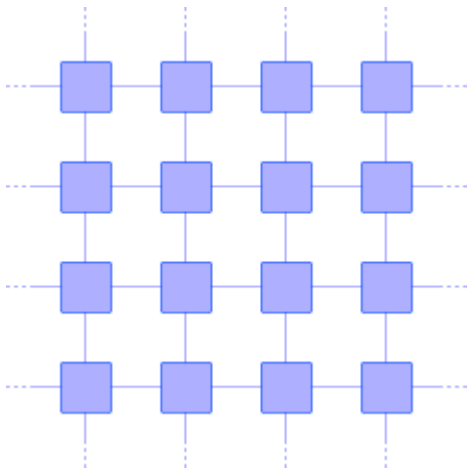
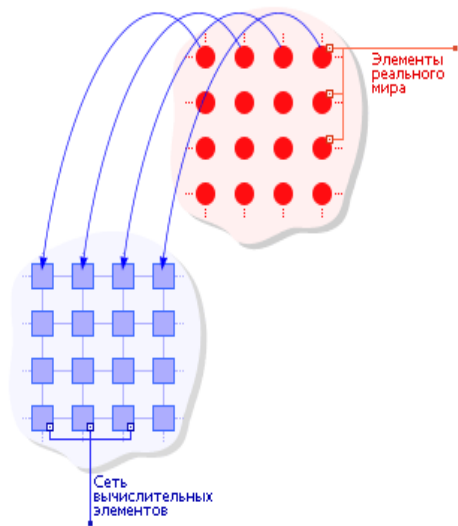


Рис. 17.4. Не фон нейманівська архітектура
обчислювальної системи (однорідна,
паралельна мережа обчислювальних
елементів)

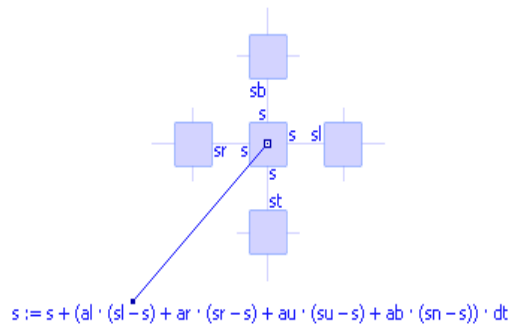
Використовуючи машину фон Неймана, архітектуру із загальною шиною (зірку), ми тим самим організуємо процес обчислень алгоритмічно, представляючи його як послідовність кроків. Сам алгоритм виконується на тому самому ресурсі - єдиному процесорі, що через шину (центральний вузол зірки) запитує крок за кроком пам'ять із приводу стану кожного з елементів, що розраховують їм. Центральний вузол (шина), таким чином, є вузьким місцем. Процесор обробляє такт за тактом кожен елемент імітованої системи, які вибудовуються в чергу до нього на обробку.

У навколишньому нас світі, що ми намагаємося імітувати, у дійсності сутності автономні, живуть як би самі по собі паралельно в часі, проявляючи свої властивості за допомогою взаємодії із сусідами по просторі.

Поставимо у взаємно однозначну відповідність елементарні сутності імітованого складного об'єкту з реального світу елементарним обчислювачам — кожен обчислювальний елемент буде імітувати одну певну сутність (див. мал. 17.5, а). Для цього в обчислювальному елементі запишемо закон функціонування саме тієї сутності, що він буде імітувати (див. мал. 17.5, б).



а)



б)

Рис. 17.5. Ілюстрація принципу об'єктного відбиття миру в інформаційних моделях

Для нашої задачі розрахунку теплопровідності запис закону зміни температури для елементарного об'єму буде мати такий вигляд:

$$s := s + (a_l \cdot (s_l - s) + a_r \cdot (s_r - s) + a_u \cdot (s_u - s) + a_b \cdot (s_n - s)) \cdot dt.$$

Кожен елемент обчислювальної мережі робить ту саму, причому, єдину, операцію, реалізує ту саму модель, причому, елементарну. Це особливість паралельних структур.

Далі зв'яжемо обчислювальні елементи зв'язками так само, як контактують між собою елементи реального миру.

Тоді елемент буде передавати відомості про свою температуру s лівому елементу в його змінну sr , правому елементу — у його змінну sl , верхньому елементу — у його змінну sb , нижньому від нього елементу — у його змінну st (див. мал. 17.5, б). at , ar , al , ab — коефіцієнти теплопровідності у відповідних напрямках. Змінна s змінюється на величину підходящих із цих напрямків теплових потоків за такт часу dt : $s := s + (...) \cdot dt$. Установивши, таким чином, зв'язку всіх елементів, ми одержимо подобу обчислювальної мережі реальному розподіленому об'єкту.

Зверніть увагу, алгоритм, показаний нами на мал. 17.2, зникає, поступаючись місцем моделі. У записі моделі ніщо не нагадує нам про специфіку середовища реалізації. Тут немає алгоритмічних структур, відповідно й немає необхідності в сторонні стосовно імітируемому об'єкта конструкціях (if, while і т.п.). Це відбувається за рахунок того, що ми використовуємо **більше повну подобу імітованої системи платформи**, на якій виробляється її моделювання. При використанні тло нейманівської структури ми змушені здійснювати особливий додатковий переклад з мови об'єкта на мову машини (процедура алгоритмізації), що не має відносини до реального об'єкта, але маючий відношення до середовища реалізації. Тому треба намагатися приділяти увагу АДЕКВАТНОСТІ платформи (архітектури, мови опису), на якій виробляється моделювання, реальному пристрою навколишні нас миру. Намагатися встановити найбільша ПОДОБА між ними ще на стадії розробки.



Отже, є дві принципово різні обчислювальні архітектури - фон нейманівська й не фон нейманівська (обидві запропоновані фон Нейманом):

- фон нейманівська - послідовна машина, зручна для опису процесів, послідовностей, що розгортаються в часі, і використовує нотацію алгоритмів, задач. Це централізована система, що реалізується за схемою «зірка»;
- не фон нейманівська - паралельна машина, зручна для опису сутностей, що використовує нотацію об'єктів, моделей. Це децентралізована архітектура, що реалізує схему «мережа».

Приклад. Промодельюємо теплопровідність будинку із двох кімнат, з'єднаних між собою дверима. У системі врахуємо можливість повідомлення кімнат з навколишнім середовищем через вікна й вхідні двері.

Запустіть послідовно три проекти: «Будинок-1», «Будинок-2» й «Будинок-3».

Зверніть увагу, у першому випадку ми розглядаємо кімнату як одне ціле. Будинок являє собою зосереджену систему із двох елементів, на які діє навколишнє

середовище [] й додаткові (зовнішні) джерела нагрівання (батареї) []. У цьому випадку розроблювач проекту припускає температуру в різних місцях однієї кімнати єдиної.

У моделі 2 проектувальник ускладнив проект, установивши в нього додаткові компоненти інтерфейсу для зручності використання. Ми бачимо процес формалізації в розвитку. На модель накладається інтерфейс, важливим тут є те, що він у даній нотації відділений від моделі. Ми можемо легко замінити окремі елементи інтерфейсу незалежно друг від друга.

Якщо ми хочемо **уточнити** модель, уважаючи, що температури в різних кутах кімнати можуть бути **різними**, що в дійсності саме так, то нам оведеться **розбити** кімнати на **більше дрібні елементи**, організувати їхній опис й установити між ними зв'язку. Імітація стане більше точною, тому в моделі 3 ви можете вже досліджувати більше тонкі ефекти — спробуйте відкривати й закривати вікна й двері, міняти температури опалювальних приладів і навколишнього середовища й подивитися на результат (див. мал. 17.6).



Рис. 17.6. Результат роботи моделі розподіленої системи при різних умовах і зовнішніх впливах

Примітка. Справедливості заради, слід зазначити, що, незважаючи на те, що ви бачите елементи обчислювальної мережі, імітуєте ви в остаточному підсумку на послідовній машині (машині процесів, алгоритмічній машині) через брак у даний момент у вас не фон нейманівської машини. Компенсує (маскує) складності алгоритмічного способу мислення середовище моделювання Stratum-2000, що імітує роботу паралельної об'єкто-орієнтованої машини на фон неймановській структурі. Але при наявності у вас не фон нейманівської машини ви б без яких-небудь змін перенесли б в автоматичному режимі проект на мережу обчислювачів. Такий перенос прийнятий називати відбиттям об'єкта в імітаційне середовище.

Тобто в проекті 3 ми фактично бачимо паралельну об'єкто-орієнтовану систему. Структура не фон нейманівської машини підтримує об'єкто-орієнтований спосіб моделювання.