

## ЗАСТОСУВАННЯ МОДЕЛЕЙ ПРОЦЕСОРІВ ТА ПРИСТРОЇВ ДЛЯ ПРОЕКТУВАННЯ ІНТЕЛЕКТУАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

©Буров С.В., 2008

**Запропоновано підхід до моделювання та проектування інтелектуальних інформаційних систем. Модель системи будується на трьох ієрархічних рівнях –бізнес-процесів, сервісів та рівні процесорів та пристроїв. Детально розглянута задача моделювання процесорів та пристроїв.**

**An approach for modeling intellectual information systems is proposed. The system is modeled on three hierarchical levels – business processes, system services and processors. The problem of modeling processors and devices within delineated approach is presented with more details.**

### Постановка проблеми та аналіз останніх досліджень

Практика проектування інформаційних систем (ІС) сьогодні виявляє недоліки та “вузькі місця”, якими характеризуються існуючі підходи та методології проектування ІС. Зокрема, існують такі проблеми:

- Проектування ґрунтується на фіксованому наборі вимог до системи, які часто визначені нечітко. Зміна вимог вимагає перепроєктування системи [1].
- Проектування найчастіше виглядає як циклічне повторення таких робіт, як Визначення вимог – > Проектування архітектури -> Придбання обладнання -> Кодування програмного забезпечення - > Тестування - > Системна інтеграція -> Впровадження та експлуатація. Реалізація подібного підходу зазвичай займає багато часу і не дає змоги гнучко реагувати на зміни середовища та вимог до системи [2].
- В основу ІС покладено програмні системи, в яких правила та алгоритми функціонування жорстко зашиті у код. Будь-яка зміна алгоритму та правил вимагає перекодування та тестування, що займає багато часу та коштів.
- Під час проектування та керування системою превалює клаптиковий підхід – підсистеми та задачі розглядають окремо одна від однієї, в результаті чого вирішення однієї проблеми в одній підсистемі створює проблеми в інших підсистемах [3].
- Керування системою (навіть прості рутинні задачі) виконується на низькому рівні з участю людей – адміністраторів, що дуже дорого та неефективно.
- Знання про систему розрізнені, часто значна кількість їх недокументована та легко втрачається [4].

Для вирішення зазначених проблем пропонуються різноманітні підходи та технології, що стосуються як архітектурних, так і методологічних принципів проектування. Зокрема, з метою збільшення ефективності розробки в умовах невизначеності та змінності вимог до системи запропоновані та успішно використовуються методології екстремального програмування (Agile)[5].

Проблема створення єдиного репозиторію даних про систему, що використовується як на етапі проектування, так і на етапі експлуатації, розглядається в концепції динамічних систем фірми Microsoft [4]. Реалізація цієї концепції знаходиться все ще на початкових етапах. Так, у продукті

MS Visual Studio реалізовано центральний репозиторій вимог до програмної системи та надана можливість встановлення зв'язків між програмними компонентами та вимогами. У серверній ОС Windows 2003 (Longhorn) додано інтелектуальні засоби для підтримки задачі розгортання інформаційної системи. У запропонованій [4] архітектурі DSI моделювання на рівні бізнес-процесів взагалі не розглядається.

### Принципи організації системи проектування

Вирішити зазначені проблеми можна в межах системи проектування, що спирається на такі принципи:

- Бізнес-процеси є джерелом вимог до інформаційної системи.
- Зміна вимог до бізнес-процесів повинна автоматично та гнучко позначатися на зміні структури та параметрів ІС, яка забезпечує функціонування бізнес-процесів.
- Застосовується модельно-орієнтований підхід до проектування та керування інформаційними системами. Моделі інкапсулюють параметри та знання про компоненти системи.
- Моделі використовують інформацію про стан модельованої системи, яка збирається в реальному масштабі часу. Це необхідно для прийняття адекватних рішень з використанням моделей.
- Поведінка системи визначається не жорстко закодованими алгоритмами, а набором моделей та правил. Зміна алгоритму роботи відбувається шляхом зміни набору моделей та правил. Моделі та правила опрацьовує спеціальний процесор, незалежний від їх змісту.
- Система проектування є інтелектуальною – використовує досвід, кращі практики та шаблони проектних рішень залежно від контексту.
- Система проектування використовує спільні для всіх підсистем онтології та моделі, що дає змогу досягти однакової трактовки та розуміння процесів та об'єктів проектування. Загалом ці онтології та моделі утворюють спільну базу знань про систему. Ці знання використовують не тільки для проектування, але й для керування під час експлуатації системи.

Система проектування будується як система багаторівневих моделей. Рівні моделювання відрізняються предметною областю, мовами моделювання, задачами та критеріями ефективності. Пропонується використовувати такі рівні (площини) (рис. 1) моделювання:

- рівень бізнес-процесів. На цьому рівні розглядають моделі бізнес-процесів та оперують поняттями, що відображають характеристики бізнес-процесів. Для побудови моделей використовують одну з мов моделювання бізнес-процесів. Тут вимірюють кількісні та оцінюють якісні параметри виконання бізнес-процесів та окремих їх складових. Оптимізують структуру та параметри процесів, оцінюють їх ефективність. Адаптують структуру та параметри БП до зміни вимог довкілля. Формулюють вимоги до інфраструктури.
- рівень сервісів. На рівні сервісів розглядають застосування та загальносистемні сервіси та ресурси, які використовують бізнес-процеси для виконання операцій. Враховують сумісне використання ресурсів, формулюють та впроваджують загальносистемні корпоративні політики, керують пріоритетністю виконання завдань
- рівень процесорів та пристроїв. Містить абстрактні моделі виконавців завдань (процесорів). Виконавцем може бути як машина певного типу, так і людина. Загальні характеристики, обмеження, моделі функціонування. Як правило, кожному процесору відповідає певний пристрій. Модель пристрою розглядають як подальшу спеціалізацію (деталізацію) моделі процесору.

Отже, інформаційна система розглядається як кортеж моделей мереж:

$$S = \{NBp, NSe, NPc\} \quad (1)$$

де  $NBp$  – мережа бізнес-процесів,  $NSe$  – мережа сервісів та застосувань,  $NPc$  – мережа процесорів.

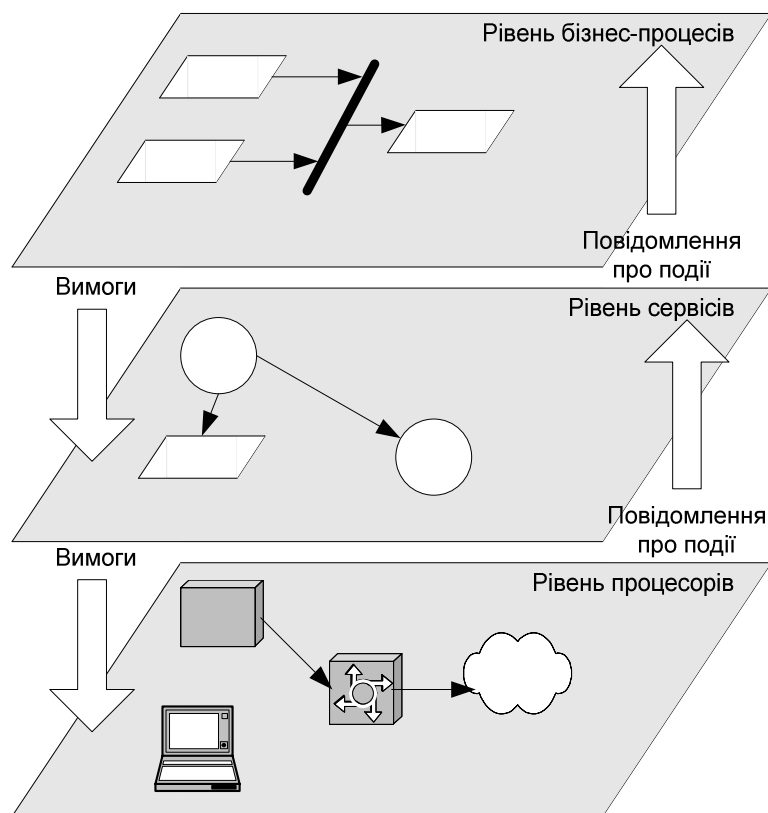


Рис. 1. Рівні моделювання

На кожному рівні моделювання розглядається комплекс взаємопов'язаних моделей, які використовують для розв'язання задач адаптації у межах цього рівня. Так, на рівні бізнес- процесів адаптувати поведінку системи можна зміною пріоритетності виконання завдань, переходом на інший варіант структури процесу. Одним з варіантів адаптаційних рішень є зміна вимог до інфраструктури. Рішення про вибір того чи іншого варіанта адаптаційної поведінки вибирається в межах рівня на основі наявної бази знань залежно від характеристики ситуації, яка вимагає адаптації системи.

Нижні рівні системи моделювання оцінюють з використанням наявних моделей можливості виконання вимоги та повідомляють вищий рівень про виконання завдання, а за необхідності ініціюють виконання цих завдань.

Моделі нижніх рівнів, своєю чергою, отримують інформацію про події у системі та повідомляють вищі модельні рівні у разі необхідності (за неможливості адаптації) для проведення необхідних адаптивних процедур вищого рівня. Наприклад, при виході з ладу одного з комп'ютерів – серверів баз даних та неможливості негайної його заміни, повідомляють рівень бізнес- процесів про тимчасову недоступність сервісу баз даних, що призупиняє проведення набору операцій, пов'язаних з цим сервісом.

У цій роботі розглянуто проблематику моделювання на найнижчому рівні запропонованої ієрархії – рівні процесорів та пристроїв.

### **Моделі процесорів та пристроїв та їх використання для моделювання сервісів інформаційної мережі**

Інтелектуальні сервіси у процесі свого функціонування використовують та спираються на об'єкти, які забезпечують функціонування сервісу, або ж створюються (споживаються) під час його функціонування. Ці об'єкти впливають на якість виконання запитів сервісом, мають власні параметри

та обмеження. Поділимо їх на два типи відповідно до складності та ступеня активності у виконанні завдань сервісу:

- процесори (виконавці) перетворюють інформацію. Вони виконують завдання, що надходять від сервісів;
- ресурси, які не виконують інформаційних перетворень.

Прикладом процесорів є комп'ютери, персонал, принтери, мережа передавання даних. Прикладами ресурсів є документи, папір для друкування.

Створювати моделі цих об'єктів необхідно для побудови інтелектуальних когнітивних систем, адже ці моделі інкапсулюють параметри та знання про відповідні об'єкти, які безпосередньо використовуються для керування процесом роботи сервісу. Нерідко процесор виконує одночасно завдання від декількох сервісів і має свої власні механізми пріоритетизації виконання. Процесор, як правило, має низку параметрів керування, метрики ефективності виконання робіт. Для кожного процесора можна побудувати моделі функціонування, які дають змогу прогнозувати та планувати роботу процесора, відслідковувати ступінь його “здоров'я”. Налаштовуючи параметри процесора, можна досягти ефективнішого виконання завдань сервісів.

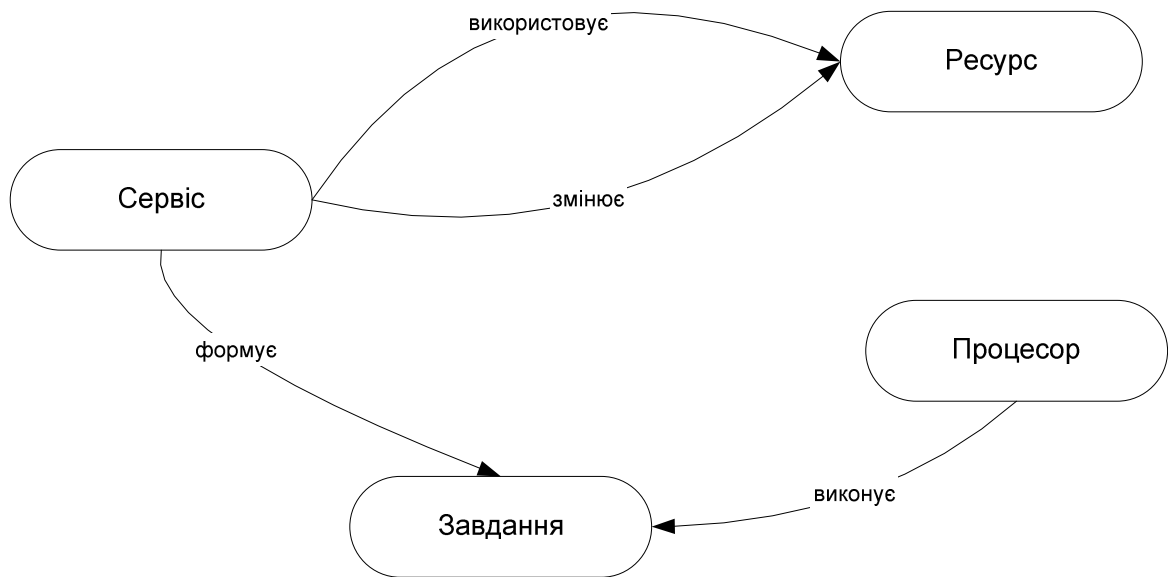


Рис. 2. Схема взаємодії сервісу та процесора

Семантична мережа (рис. 2) ілюструє відмінність понять “процесор” та “ресурс”. Як правило, сервіс формулює для процесора певне завдання для виконання. Взаємодія ж сервісу та ресурсу найчастіше визначається такими дієсловами, як використовувати, змінювати, відсилати, отримувати, створювати (знищувати) тощо.

Процесор та ресурс відрізняються від сервісу тим, що сервіс є абстракцією, яка визначає процес виконання запитів, а процесор (ресурс) є абстракцією реального об'єкта (комп'ютера, програми, документа). Один і той самий процесор може одночасно виконувати завдання багатьох сервісів.

З іншого боку, виникає запитання: чим відрізняються моделі процесорів від моделей пристроїв? Модель процесора – це узагальнена модель, спільна для групи пристроїв. Така модель необхідна на етапі проектування, коли конкретні пристрої та їх параметри ще не відомі. Навіть якщо реальна система вже існує, то використання узагальненої, абстрактної моделі виправдано скороченням коштів на розроблення моделей для пристроїв. У загальному випадку модель пристрою, подана як клас об'єктно-орієнтованого проектування, належить до моделі процесора як клас-спеціалізація – до базового класу. Вона успадковує всі дані та методи від базового класу, та, можливо, до визначає (або перевизначає) якісь параметри та методи базового класу.

### Задачі, які вирішуються з використанням моделей процесорів

В інтелектуальній когнітивній системі моделі процесорів створюють для вирішення таких завдань:

- відслідковування стану системи, адаптація системи до зміни стану, самолікування та самопоновлення. Вирішення цієї задачі передбачає наявність процесу моніторингу параметрів процесора, моделей “здоров’я” системи, які дають змогу приймати рішення про дії у випадку виникнення визначених відхилень параметрів.
- адаптація параметрів системи для виконання визначених завдань, забезпечення якості виконання завдань.
- планування виконання завдань у системі, особливо якщо процесор виконує одночасно декілька завдань.
- прогнозування виконання завдань. Визначення очікуваних параметрів якості виконання визначеного завдання.
- аналіз роботи за визначений період, збирання статистики виконання завдань, коригування моделей, що використовуються для планування та прогнозування за реальними даними.

Отже, модель процесора – це комп’ютерна модель, яка відображає структуру, стан, можливості та обмеження процесора з виконання завдань. Ця модель виконується у реальному масштабі часу та інкапсулює знання про об’єкт моделювання.

### Структура інтелектуальної системи моделювання

Комплекс задач, які вирішує система моделювання рівня процесорів та пристроїв, передбачає таку послідовність роботи системи моделювання (рис. 3).

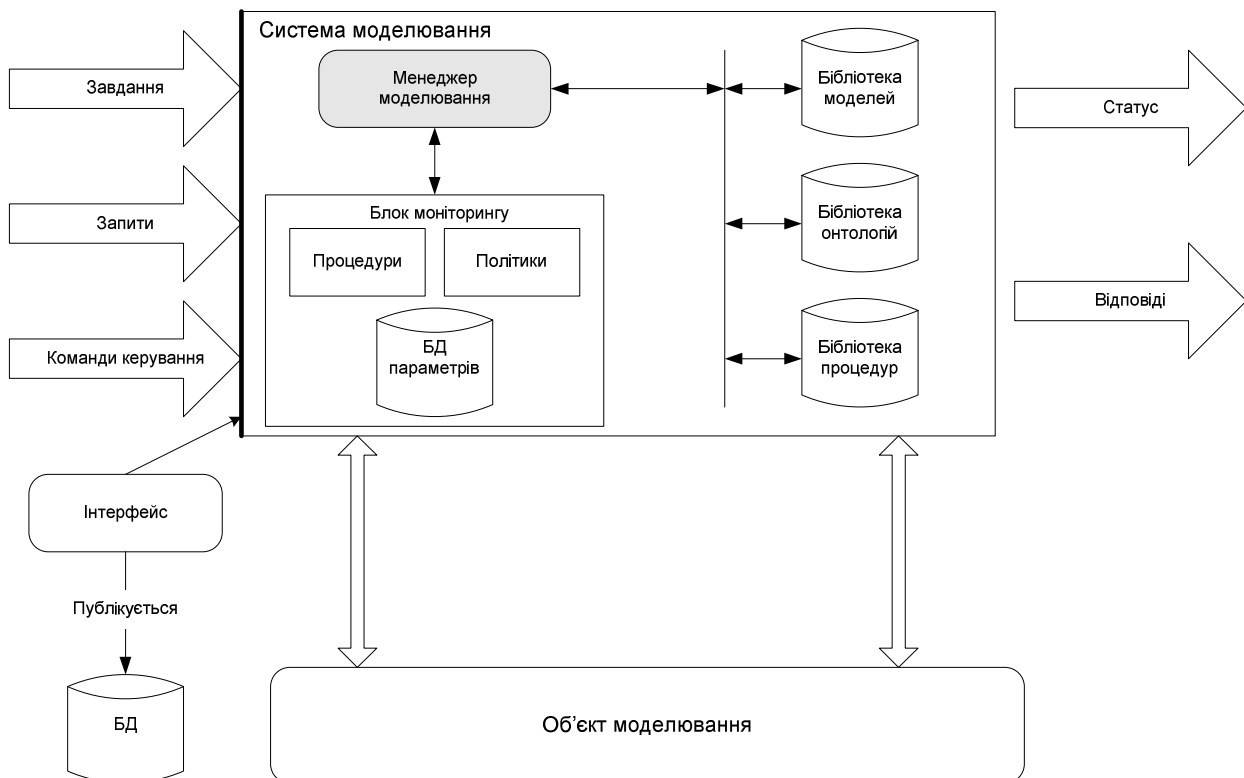


Рис. 3. Структура системи моделювання

На вхід системи моделювання надходять запити, завдання на виконання, команди керування. На виході системи є відповіді на запити, статусні повідомлення. Система моделювання пов’язана також і з

об'єктом моделювання, передаючи йому завдання на виконання, збираючи значення параметрів моніторингу, змінюючи параметри налаштувань та виконуючи процедури керування та підтримки.

Весь перелік можливих вхідних та вихідних повідомлень формалізується як інтерфейс та публікується (наприклад, з використанням протоколу UDDI), так що зовнішні програми, що звертаються до системи, не потребують жорсткої прив'язки до інтерфейса доступу. Це дає змогу змінювати набір послуг та моделей, які використовуються системою моделювання.

Менеджер моделювання опрацьовує зовнішні запити та завдання, обирає потрібні моделі для виконання. Алгоритми його роботи сформульовані у бібліотеці моделей.

Бібліотека моделей містить різноманітні моделі, що використовуються для генерації відповідей на зовнішні запити, адаптації до змін. Прикладом таких моделей є моделі продуктивності, моделі здоров'я, моделі планування, прогнозування тощо. Варто зазначити, що вживані моделі відрізняються ще й рівнем загальності та деталізацією. Наприклад, можуть існувати різні моделі, що мають однакові інтерфейси: модель, що розглядає процесор на рівні його типу, і модель, що враховує особливості конкретного пристрою.

Бібліотека онтологій необхідна для підтримки спільного розуміння концептів моделювання та прийняття рішень на основі семантичних даних.

Бібліотека процедур містить процедури багатаразового використання, які застосовуються при взаємодії з об'єктом керування.

Блок моніторингу містить процедури, політики та базу даних моніторингу. Завданням блоку моніторингу є збирання, збереження даних про стан об'єкта моделювання. Дані моніторингу зберігають у базі даних та використовують у моделях для прийняття рішень.

Політики моніторингу визначають, які параметри і як часто треба моніторити. Бібліотека процедур моніторингу містить код керування та моніторингу.

#### **Формальна специфікація системи проектування та моделювання**

Отже, мережа пристроїв – це множина моделей пристроїв (ресурсів, процесорів):

$$NPc = M(Pc).$$

Кожна така модель подана як опис інтерфейсу та реалізації:

$$Pc = \{InPc, RePc\}.$$

Інтерфейс визначається як множина команд (та відповідей), які публікуються

$$InPc = M(InCm).$$

Реалізація запропонованої системи моделювання процесорів має такі складові частини:

$$RePc = \{CnPc, LiMoPc, LiOnPc, LiCdPc, MnPc\},$$

де  $CnPc$  – менеджер моделювання,  $LiMoPc$  – бібліотека моделей,  $LiOnPc$  – бібліотека онтологій,  $LiCdPc$  – бібліотека процедур,  $MnPc$  – блок моніторингу. Блок моніторингу складається з бази даних параметрів  $PmMnPc$ , бібліотеки процедур  $LiCdMnPc$  та набору політик моніторингу  $PlMnPc$ :

$$MnPc = \{PmMnPc, LiCdMnPc, PlMnPc\}.$$

#### **Порівняння з існуючими аналогічними розробками**

Запропонована архітектура системи моделювання спрямована на вирішення задач адаптації, самоналаштування, самолікування та подібна функціонально до архітектури автономних обчислень IBM[5], когнітивної системи FOCALE[6] та концепції динамічних систем фірми Microsoft [4].

Водночас запропоновані вирішення мають ряд переваг:

- система моделювання подана як сервіс, що дає змогу змінювати як реалізацію системних функцій, так і її інтерфейс, а також спрощує доступ до неї різних клієнтів з різних рівнів моделювання;
- порівняно з [5] та [4] система доповнена функціонально. Так, вона підтримує запити про стан системи, аналіз та прогнозування роботи керованої системи;
- на відміну від системи FOCALÉ [6] адаптація здійснюється не тільки за відхиленням, але й з використанням довільних критеріїв, відображених у моделях;
- для прийняття рішень та відповідей на запити використовується багато моделей, кожна з яких відображає певний аспект системи (наприклад, продуктивність, архітектуру, здоров'я та ін.

### Висновки

Запропонований у роботі підхід до проектування та моделювання інтелектуальних інформаційних систем дасть змогу створювати системи, які порівняно з існуючими швидше реагують на зміни вимог та зовнішнього середовища.

1. Balmelli L, Brown D, Cantor M, Mott M. *Model-driven systems development [Text]* / Balmelli L, Brown D, Cantor M, Mott M // *IBM Systems Journal*. – 2006. – Vol. 45, # 3. 2. Jason Charvat. *Project management methodologies. [Text]*. – Charvat J.-John Wiley&Sons. – 2003. – ISBN 0-471-22178-3. 3. Вячеслав Ерохин. *Инструменты управления архитектурой предприятия [Текст]*/ Ерохин В// *Открытые системы*. – 2007. – № 03 4. *Microsoft Dynamic Systems Initiative Overview [Electronic resource]*. – Microsoft Windows server system white paper. – Microsoft Corporation. – March 2005 Mode of access: URI: <http://www.microsoft.com/business/dsi/dsiwp.msp>. 5. Henrik Kniberg. *Scrum and XP from trenches [Electronic resource]*/Kniberg H. – *Crisp white paper*. – version 2.2 2007-04-21. – Mode of access: URI:[www.crisp.se](http://www.crisp.se) 6. Джеффри Кефарт, Дэвид. *Концепция саморегулирующихся вычислений [Текст]* / Кефарт Д., Чесс Д. *Открытые системы*, #02, 2003 7. John Strassner. *The Role of Autonomic Networking in Cognitive Networks [Text]* / *Cognitive networks. Towards self-aware networks. Edited by Qusay H. Mahmoud University of Guelph, Canada. John Wiley & Sons*. – 2007.