

УДК 681.518:681.327.8

Є.В.Буров

НУ "Львівська політехніка", кафедра інформаційних систем та мереж

СИСТЕМА ФОРМАЛЬНИХ СПЕЦИФІКАЦІЙ ДЛЯ ПРОЕКТУВАННЯ РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

© *Є.В.Буров, 2000*

Actual acceleration of business processes modification presents new challenges to information system automation design tools and concepts. This paper propose an methodology and formal specification system to assist information design process during complete life-cycle with special emphasis on physical design stage.

Сучасний етап розвитку мережевих інформаційних технологій відрізняється рядом тенденцій, які вимагають наукового осмислення та нових системотехнічних вирішень. Насамперед це зростання складності інформаційних систем. Сучасні корпоративні мережі нерідко нараховують сотні серверів та десятки тисяч робочих станцій — клієнтів, що працюють одночасно з багатьма застосуваннями. Складність систем породжує нові проблеми у керуванні, діагностиці, експлуатації та адмініструванні систем. Розвиток мультимедійних технологій покликав до життя нові технології контролю та забезпечення якості передавання. У ширшому контексті це створює можливості керування параметрами якості роботи застосувань та виконання бізнес-процесів підприємства.

Паралельно з розвитком інформаційних технологій істотні зміни відбуваються і у сфері керування виробництвом. Довший час інформаційні системи та методи їх проектування розвивалися незалежно від методів удосконалення організаційно-виробничих процесів.

Із середини 80-х років ринкова ситуація докорінно змінилася. Головними причинами таких змін були зростання доступності товарів та послуг виробників з усього світу та викликане цим збільшення вимог користувачів щодо їх якості та різноманітності, що призвело до різкого зростання конкуренції в плані пропозиції нових товарів, підвищення їх якості, а також зменшило час життя товару і послуги на ринку.

Водночас зі збільшенням динаміки ринку товарів та послуг збільшується складність нових продуктів, їх наукоємність. Певний час в розвинених економіках [1] це призводило до збільшення кількості менеджерів середньої ланки. Зважаючи на великі пов'язані з цим витрати, сьогодні цей резерв розвитку практично вичерпано. Ці причини, а також недостатня ефективність роботи існуючих інформаційних систем призвели до появи нових підходів щодо реорганізації діяльності підприємств — реінжинірингу бізнес-процесів (BPR — Business Process Reengineering) та нового системного проектування (НСП) як поєднання інформаційних технологій, BPR та соціально-психологічних методів організації праці.

Безпосереднім провідником сучасних інформаційних технологій є інформаційна система (ІС) підприємства. Більшість сучасних ІС використовують комп'ютерні мережі, тому й не дивно, що їх інколи ототожнюють з розподіленими інформаційними системами (РІС).

Збільшення динаміки змін бізнес-процесів підприємств ставить додаткові вимоги до РІС, передусім у напрямках збільшення ступеня їх адаптивності, керованості, простоти перенастроювання, зменшення часу проектування та перепроектування. Розв'язують цю проблему завдяки стандартизації та уніфікації компонент ІС, повторному використанню програмних та технічних компонент, застосуванню принципів архітектурного підходу до проектування та розвитку РІС.

Одним з важливих напрямів розв'язання проблеми швидкої адаптації РІС до змін бізнес-процесів є розробка систем автоматизованої підтримки процесів проектування, модифікації, моделювання, керування та налаштування ІС.

Наявні CASE-засоби автоматизації проектування ІС головним чином відображають задачі логічного проектування, тоді як фізичне проектування намічене тільки контурно. Існує певний розрив між задачами та засобами логічного та фізичного проектування.

Незважаючи на численні роботи, присвячені різним задачам фізичного проектування, загалом вони не інтегровані навколо єдиних даних проектування, що робить їх використання утрудненим.

Для розв'язання наведених вище задач інструментальна система автоматизації проектування РІС на фізичному рівні мусить мати такі властивості:

- підтримувати наявні методології в частині логічного проектування;
- базуватися на системі формальних специфікацій, що забезпечить дотримання єдиних архітектурних принципів, полегшує доповнення та модифікацію системи;
- підтримувати задачі всього ЖЦ системи, зокрема і задачі керування та налаштування;
- давати змогу проводити проектування як 'зверху-вниз', так і 'знизу-вверх';
- надати можливість використовувати наявні методи аналізу та оптимізації в межах єдиної системи, єдиної моделі даних;
- підтримувати автоматизацію, тобто програмні модулі системи повинні видавати дані на запит не тільки людині-проектувальнику, але й програм для генерування оцінок та рекомендацій;
- використати традиційні переваги CASE-систем, які полягають в інтеграції процедур, багаторазовому використанні результатів, інтеграції даних проектування у репозиторії, колективному використанні інформації, нелінійному процесі розробки.

Центральною частиною системи проектування є система формальних специфікацій проектування.

Система загалом специфікується чотирма мережами:

$$S = \{NPc, NPr, NPt, NDv\},$$

де NPc — мережа процесів, NPr — мережа процесорів, NPt — мережа прототипів, NDv — мережа пристроїв.

На ранніх етапах проектування під процесом розуміють бізнес-процес, певну процедуру, зрозумілу Замовнику. На подальших етапах проектування процес — це операція, яка супроводжується перетворенням інформації (матеріальних потоків, параметрів системи).

Мережа процесорів NPr є подальшим кроком у побудові специфікації системи. Специфікація ускладнюється за рахунок розкриття структури розподілених процесів, визначення та ідентифікації компонент багаторазового використання — системних сервісів, а також визначення та опису об'єктів-виконавців процесів (процесорів). На цьому етапі приймають загальноархітектурні рішення.

На етапі побудови мережі прототипів NPt кожному процесору ставиться в відповідність певний прототип пристрою, або технічного вирішення. При цьому використовують прості експертні системи.

Будуючи мережу пристроїв NDv , для кожного прототипу визначають марки та моделі реальних пристроїв.

Розглянемо кожен тип специфікаційної мережі детальніше.

Мережа процесів NPc — це множини процесів та міжпроцесних потоків.

$$NPc = \{M(Pc), M(LPc)\}$$

де $M(Pc)$ — множина процесів Pc , $M(LPc)$ — множина потоків LPc між ними.

Процес

$$Pc = \{idPc, TyPc, PbPc, DtPc, MdPc, CmPc\}$$

де $idNPc$ — ідентифікатор процесу, $TyPc$ — тип процесу, $PbPc$ — блок параметрів, $DtPc$ — деталізація процесу, $MdPc$ — параметрична модель виконання, $CmPc$ — коментар.

Ідентифікатор процесу однозначно визначає реалізацію процесу. Тип процесу специфікує набір параметрів процесу та операції над ним. Тип належить до ієрархії типів процесів. Ієрархія типів процесів є одним з різновидів метаданих проектування (інструментальних даних, які використовуються у багатьох проектах) та зберігається у Словнику даних (data dictionary).

Блок параметрів процесу

$$PbPc = \{PrPc, MePc, ZaPc, CmPc\} \quad (1)$$

де $PrPc$ — множина параметрів, $MePc$ — методи роботи з параметрами, $ZaPc$ — параметричні залежності, $CmPc$ — коментар.

Параметричні залежності та параметри можуть бути невизначеними. Взагалі кожен параметр може:

- мати конкретне значення;

- належати до певної множини значень;
- бути невизначеним;

Можливість невизначеного значення параметра дає змогу довизначити його потім, відкинути існуюче значення, якщо процедура визначення параметра виявиться недостовірною, запрограмувати реакцію системи на невизначеність параметра, передбачити його визначення в результаті складних, багатоетапних визначень інших параметрів через параметричні залежності.

Можливість приналежності параметра до множини значень дає змогу працювати з нечіткими значеннями та логіками, визначати області коректної роботи системи у вигляді параметричних інтервалів (або n -мірних доменів), розглядати узагальнений параметр як область n -мірного простору, працювати з множинами з безмежною кількістю елементів.

Параметрична залежність — це функція, яка дає змогу оцінити значення визначеного параметра за значеннями інших параметрів. Крім функції, опис залежності містить ідентифікатор залежності, коментар, ідентифікатор параметра, який визначається. Визначеною будемо вважати таку параметричну залежність, для якої існують всі її параметри (при цьому значення деяких параметрів може бути невизначеним).

Частиною кожного процесу може бути його деталізація. Деталізація — це структура процесу.

Деталізація

$$DtPc = \{idDtPc, NpDtPc, LDPc, PbDtPc, CmDtPc\},$$

де $idDtPc$ — ідентифікатор, $PcDtPc$ — множина процесів, що входять у деталізацію, $LDPc$ — множина потоків між процесами, $PbDtPc$ — блок параметрів, $CmDtPc$ — коментар.

Деталізація обмежена щодо параметрів. Вона успадковує параметри свого типу процесу та батьківського процесу.

Модель виконання має на меті подати в наочній формі механізм функціонування процесу з урахуванням його параметрів та параметрів вхідного потоку.

Є два підходи до подання та розгляду параметричної моделі виконання.

В першому на кожен момент часу вважається, що вхідні потоки, стан та вихідні потоки мають постійні параметри (система стаціонарна). Вхідний потік та вихідний потік вважається неперервними, стаціонарними.

Зміна параметрів вхідного потоку, параметрів керування процесом призводить до зміни параметрів вихідного потоку та стану. Вважається, що після певного перехідного процесу всі параметри знов набудуть стаціонарних значень (зміниться стан системи).

Метою такої моделі виконання є визначити параметри стану та вихідного потоку при даному стані та вхідному потоці.

Моделі першого типу краще підходять для стадії роботи з мережею процесів, якщо:

- на цій стадії потоки неявно розглядаються неперервними,
- параметри потоків та стану набувають неперервних значень, для яких неможливо

- говорити про якийсь дискретний вхідний алфавіт;
- розгляд механізму переходів між станами належить до реалізаційної компоненти і на цьому етапі недоречний.
- потокові моделі розглядають процес як чорну скриньку, а для опису параметричних залежностей використовуються прості залежності або моделі нижчих рівнів.

Модель виконання містить такі функції:

параметричну функцію виходу

$$F1: PbPc \times Pb_inLPc \rightarrow Pb_oLPc,$$

параметричну функцію зміни параметрів

$$F2: PbPc \times Pb_inLPc \rightarrow PbPc$$

Модель другого типу доцільно розглядати, якщо процес працює в режимі запит-відповідь (наприклад, розглядається проходження обслуговування однієї заявки клієнта). У такому разі для його опису можна зупинитися на одному з існуючих засобів специфікації. Такими засобами можуть бути скінченні автомати, мережі Петрі, ESTELLE, Lotos, інші мови формальної специфікації. Більшість з цих засобів оперує станами системи та є розширеннями скінчених автоматів (ESTELLE, AMN).

Зв'язок процесів (потік).

Під час розгляду потоку LPc припускається, що він не виконує перетворення інформації (якщо таке перетворення є, вводиться додатковий процес транспортування). Тому потік не потребує координації та ресурсів.

$$LPc = \{ idLPc, id_inPc, id_ouPc, PbLPc, DtLPc, CmLPc \},$$

де $idLPc$ — ідентифікатор потоку, id_inPc — ідентифікатор процесу — джерела потоку, id_ouPc — ідентифікатор процесу — споживача потоку, $PbLPc$ — параметрична компонента, $DtLPc$ — деталізація, $CmLPc$ — коментар.

Деталізація $DtLNP$ передбачає тільки одну операцію — розбиття на множину потоків, дослідження та моделювання взаємозв'язку потоків. Цей взаємозв'язок моделюється параметричними залежностями.

На етапі побудови мережі процесів вирішують такі задачі:

- задачі, пов'язані з розробкою самої системи специфікацій (деталізація, агрегування).
- задачі, пов'язані з параметричним аналізом та параметричними залежностями. (дослідження параметричних залежностей, доведення параметричної достатності, доведення параметричної стійкості, параметрична верифікація);
- прикладні задачі (оцінка часових параметрів, аналіз відстаней та розміщень процесів, аналіз структури потоків).

Мережа процесорів — це множина процесорів та зв'язків між ними.

$$NPr = \{M(Pr), M(LPr)\}$$

Процесор Pr — це елемент специфікації, що агрегує процеси та є носієм параметрів виконання процесів.

$$Pr = \{ id Pr, TyPr, L Pr, Co Pr, M(PcPr), idPr Pr, MdPr, Pb Pr, Dt Pr, Cm Pr \}$$

де $id Pr$ — ідентифікатор процесора, $TyPr$ — тип процесора, LPr — множина потоків керування процесора, $CoPr$ — процес керування процесора, $M(PcPr)$ — множина процесів, що виконуються цим процесором, $idPrPr$ — ідентифікатор процесору, що містить цей процесор (якщо процесор вкладений в інший), $MdPr$ — параметрична модель виконання, $PbPr$ — блок параметрів процесора, $DtPr$ — деталізація процесора, $CmPr$ — коментар.

Типізація процесора визначає структуру його параметрів та модель виконання. Наприклад, можливі такі базові типи процесорів, як організація, підрозділ, комп'ютер, друкарка, виконавець — людина, комунікаційна мережа. Типи процесорів утворюють ієрархію типів, що зберігається у Словнику даних.

Процес керування $Co Pr$ визначає логіку опрацювання процесів. Він створює нові процеси, присвоює їм пріоритети, визначає інші дані, що впливають на виконання процесів, опрацьовує ситуацію закінчення процесів та ін. При цьому він працює з параметрами керування (власними та процесора).

Модель виконання $MdPr$ специфікує опрацювання процесом керування робочих процесів. Можна сказати, що для процесу керування подією є активізація потоку до якогось з передбачених, закріплених за цим процесором процесів. У такому разі процес керування звертається до бібліотеки шаблонів, в якій записана модель виконання кожного типу процесу. Для повного опису такої моделі достатньо визначити для кожного робочого процесу $PcPr_i \in Pr$ процедуру φ_i .

$$\varphi PcPri = \{\varphi PcPri_1, \varphi PcPri_2\}$$

У найпростішому випадку опрацювання процесу складається з опрацювання двох подій — початку процесу та його закінчення. Процедура φ_i визначає дії процесу керування залежно від параметрів самого процесора, вхідного потоку робочого процесу на початку опрацювання процесу, а φ_{i2} — на закінчення. Фактично ці функції — це бізнес-правила опрацювання процесу залежно від поточних параметрів самого процесу та процесора. Отже, процес керування розглядає роботу процесора як систему моделювання за подіями.

Якщо спрощена модель робочої операції у вигляді подій початку та закінчення не задовольняє проектувальника, може розглядатися деталізація $DtPr$, яка точніше описує перебіг виконання процесу з точки зору моделювання подій та зміни параметрів.

У параметричному блоці містяться параметри процесора. Він складається з множин параметрів та методів і параметричних залежностей. Його структура аналогічна до структури блока параметрів процесу (1).

Крім процесорів, на етапі побудови мережі процесорів розглядають й інші об'єднання процесів, такі як сервіси, логічні зони, розподілені процеси.

Сервіс — це агрегат процесів з фіксованим інтерфейсом, логічна компонента введення якої спрямоване на виділення та оформлення в загальному наборі функцій однакових або споріднених функцій з метою їх подальшої реалізації у компонентах (логічних, програмних та апаратних) багаторазового використання. Такий підхід дає наступні переваги:

- здешевлює розробку;
- робить розробку надійнішою, оскільки інкапсуляція функцій в межах компоненти дає змогу краще протестувати його;
- робить систему більш продуктивною при однакових витратах завдяки спеціалізації компоненти;
- система, орієнтована на сервіси, гнучкіша до змін – нема прив'язки до конкретної структури бізнес-процесів;
- використання стандартних компонент та сервісів дає змогу застосувати динамічну реплікацію, дублювання з меншими витратами;
- відділення інтерфейсу від реалізації дає змогу змінювати реалізацію без зміни інтерфейсу.

Не дивно, що у багатьох роботах з системного аналізу та проектування ПІС підкреслюється зручність, необхідність сервісного підходу до проектування ПІС [2].

У специфікації сервісу необхідно розділити інтерфейсну та реалізаційну частини і забезпечити їх незалежні опис та зміни. Отже, сервіс:

$$Se = \{idSe, inSe, reSe, CmSe\},$$

де *idSe* — ідентифікатор сервісу, *inSe* — інтерфейсна частина, *reSe* — реалізаційна частина, *CmSe* — коментар.

Розподілений процес — це процес, який виконується на декількох процесорах і, отже, не має однозначної локалізації. Часто (але не завжди) розподілений процес є *сеансовим* процесом. В сеансовому процесі інформація про нього зберігається в учасників сеансу та використовується для керування ним. Різновидом сеансового процесу є *транзакція* — розподілений процес, який має властивість збереження цілісності.

Кожний розподілений процес є різновидом звичайного процесу. Це означає, що він має початок та закінчення, описує перетворення вхідного потоку даних у вихідний. Водночас деталізація такого процесу розносить його підпроцеси по різних процесорах та описує взаємодію з процесом керування кожного процесора.

Логічна зона — це іменоване об'єднання процесів.

$$LzPr = \{idLzPr, LiPcLzPr, TyLzPr, CpLzPr, CmLzPr\},$$

де *idLzPr* — ідентифікатор логічної зони, *LiPcLzPr* — список процесів, *TyLzPr* — тип спільності процесів, об'єднаних у зону, *CpLzPr* — властивості, які характеризують спільність, *CmLzPr* — коментар.

Переважно, у логічну зону об'єднують процеси, які мають спільні ознаки. Такими ознаками, наприклад, можуть бути: приналежність організації або підрозділу, знаходження в одному будинку, сильний міжпроцесний інформаційний зв'язок, робота з одним провайдером.

Типи спільності процесів утворюють попередньо визначену ієрархію. Елементи ієрархії визначаються задачами, які використовують логічні зони як вхідні дані.

Міжпроцесорні потоки LPr реалізують ідею керування параметрами процесора. Зв'язок (потік) керування LPr або LPc (залежно від того, хто виступає приймачем потоку — процесор або процес) — це команда зміни керуючих параметрів процесора або процесу, які впливають на їх поведінку із виконання головних функцій.

Рішення про зміну параметрів приймає процес керування старшого (охоплюючого) процесора у випадку адаптивної системи або проектувальник, системний аналітик.

$$LPr = \{ idLPr, id_inLPr, oNpNs, PbLNs, DtLNs, KLNs \},$$

де $idLPr$ — ідентифікатор потоку, id_inLPr — ідентифікатор джерела потоку, id_ouLPr — ідентифікатор приймача потоку, $PbLPr$ — блок параметрів потоку, $DtLPr$ — деталізація потоку, $CmLPr$ — коментар.

Задачі, що розв'язують на етапі побудови мережі процесорів, можна поділити на такі групи:

- деталізація мережі процесів та розбудова мережі процесорів (декомпозиція процесів, групування процесів, визначення та ідентифікація процесорів),
- прийняття рішень щодо архітектури системи,
- задачі моделювання та дослідження параметричних залежностей (ідентифікація, аналіз параметричної моделі виконання та параметричних залежностей, аналіз впливу параметрів налаштування)
- прикладні задачі (аналіз та моделювання структури подій, керування користувачами, інформаційними потоками, проектування комунікаційної підсистеми).

Головним завданням етапу побудови мережі прототипів є поставити у відповідність процесорам деяких типів прототипи пристроїв. Мережа прототипів у такий спосіб доповнює специфікації проектування новим специфікаційним рівнем.

Мережа прототипів — це множина прототипів.

$$NPt = \{ M(Pt) \}$$

Кожен прототип

$$Pt = \{ idPt, TyPt, idPr, PbPt, MdPt, CmPt \},$$

де $idPt$ — ідентифікатор прототипу, $TyPt$ — тип прототипу, $idPr$ — ідентифікатор процесора, $PbPt$ — блок параметрів прототипу, $MdPt$ — модель прототипу, $CmPt$ — коментар.

Тип прототипу належить до одного з типів ієрархії прототипів $IrPt$. Блок параметрів $PbPt$ зберігає параметри конкретного прототипу, модель $MdPt$ визначає параметричну поведінку прототипу.

Ієрархії прототипів $M(IrPt)$ зберігаються у Словнику даних. Кожна з ієрархій відповідає одному з типів процесора та має спільну процедуру вибору прототипу з множини прототипів, визначеної ієрархією.

Ієрархія прототипів

$$IrPt = \{idIrPt, TyPr, M(sPt), ChIrPt, CmIrPt\}$$

де $idIrPt$ — ідентифікатор ієрархії, $TyPr$ — тип процесора, $M(sPt)$ — множина шаблонів прототипів sPt , які в сукупності визначають всі можливі варіанти реалізації процесора, $ChIrPt$ — процедура вибору прототипу, $CmIrPt$ — коментар.

Шаблон прототипу специфікує певний варіант вибору.

$$sPt = \{id_sPt, Pb_sPt, CmPt\},$$

де id_sPt — ідентифікатор шаблону, Pb_sPt — блок параметрів, $CmPt$ — коментар.

Найважливішою частиною специфікації шаблону є його блок параметрів. Він визначає параметричні вимоги придатності прототипу.

Задача побудови мережі прототипів вирішується з застосуванням процедури вибору та параметричних визначень прототипу. Процедура вибору розглядає прототипи визначеного ієрархічного рівня в цілому. Вона може бути подана у вигляді дерева вибору. Під час розгляду такого дерева система може формулювати додаткові питання до проектувальника або до параметрів специфікаційних об'єктів.

Через те, що окремі вирішення не є повністю незалежними, пропонується визначати прототипи у такій послідовності:

- серверне вирішення;
- клієнтська частина;
- периферія;
- комунікаційна мережа;
- серверна ОС;
- СУБД, інші серверні застосування;
- клієнтські ОС;
- клієнтські застосування.

При цьому береться до уваги, що загальноархітектурні вирішення розробляють на етапі побудови мережі процесорів.

Мережа пристроїв NDv — це специфікація реальних пристроїв. Кожному з прототипів ставиться у відповідність реальний пристрій та його параметри.

Отже, мережа пристроїв — це множина специфікацій пристроїв Dv .

$$NDv = \{M(Dv)\}$$

$$Dv = \{idDv, TyDv, PbDv, MdDv, CmDv\}$$

де $idDv$ — ідентифікатор пристрою, $TyDv$ — тип пристрою, $PbDv$ — блок параметрів, $MdDv$ — модель пристрою, $CmDv$ — коментар.

Тип пристрою належить до ієрархії типів пристроїв, яка зберігається у Словнику даних. Пристрої одного типу взаємозамінні. Модель пристрою $MdDv$ відображає його функціонування.

Процес вибору має діалоговий характер. Система пропонує конкретні марки пристроїв, які за своїми параметрами відповідають параметрам прототипу. При цьому враховується сумісність та додаткові властивості наборів пристроїв.

Для повністю сформованої специфікації визначається сумарна вартість обладнання. Запропонована система специфікацій може бути покладена в основу інструментального засобу проектування розподіленої інформаційної системи.

1. Зиндер Е.З. *Новое системное проектирование: информационные технологии и бизнес-реинжиниринг: часть 2* //СУБД, 96, №1. 2. Ладыженский Г.М. *Архитектура корпоративных информационных систем.* //СУБД, 97, №5-6.

УДК 681.3

О.М.Верес

НУ "Львівська політехніка",

сектор опрацювання даних кадрового складу та зрізів знань студентів

ПОБУДОВА МНОЖИНИ КРИТЕРІЇВ ОПТИМІЗАЦІЇ УКЛАДАННЯ РОЗКЛАДУ НАВЧАЛЬНИХ ЗАНЯТЬ У ВЗО

© *О.М.Верес, 2000*

Such terms as "disition", "purpose", "criteria", "indicators" are described in this article. Grounded set of criteria optimisation arangement of establishing the studing process scheduling.

Підвищення якості рішень, які приймає керівник, — особа, що приймає рішення (ОПР), загалом є важливим резервом підвищення ефективності суспільного виробництва.

Поняття "рішення" в науковій літературі трактується по-різному. Воно розуміється і як процес, і як акт вибору, і як результат вибору [1]. Основна причина неоднозначності тлумачення поняття "рішення" полягає в тому, що кожний раз в це поняття вкладається зміст, який відповідає конкретному напрямку досліджень.

Рішення як процес характеризується тим, що він у часі проходить в декілька етапів. У зв'язку з тим можна говорити про етапи підготовки, прийняття і реалізації рішення. Етап прийняття рішення можна тлумачити як акт вибору, який здійснюється індивідуально або групою осіб, яка приймає рішення (ОПР) за допомогою визначених правил.