

ЗАСТОСУВАННЯ ІМІТАЦІЙНИХ МОДЕЛЕЙ ДЛЯ УПРАВЛІННЯ РИЗИКАМИ ІТ-ПРОЕКТІВ

© Рішняк І.В., 2010

Розглянуто імітаційні моделі аналізу ризиків ІТ-проектів та запропоновано інформаційну систему для програмної реалізації цих моделей.

Ключові слова: ризик, ІТ-проект, імітаційна модель.

The article deals with the simulation models for the risk analysis of IT-projects and an information system for software implementation of these models.

Keywords: risk, IT-project, simulation model.

Постановка проблеми та її зв'язок з практичними завданнями

Одним з головних завдань, які вирішують у межах управління проектами, є управління ризиками проектної діяльності, або управління ризиками проекту. Це завдання не відокремлюється від більшості інших функцій управління проектами. Під час визначення фінансових потреб, обчислення кошторису і бюджету, підготовки та укладання контрактів, під час контролю за реалізацією проекту постає завдання захисту учасників проектної діяльності від різних видів ризиків.

Ризики виникають на всіх етапах проектної діяльності, тому функція управління ними є актуальною аж до закриття проекту. Управління проектними ризиками "пронизує" всі без винятку напрями діяльності в межах управління проектами. Це зумовлює виникнення різних труднощів (організаційних, кадрових, психологічних тощо) щодо виокремлення цієї функції в самостійний елемент організаційної структури управління проектами.

Аналіз останніх досліджень

У межах управління ризиками ІТ-проектів актуальними є такі питання:

- класифікація проектних ризиків;
- методи виявлення й оцінювання ризиків;
- інформаційне забезпечення управління ризиками;
- моніторинг і прогнозування ризиків;
- технології зниження ризиків;
- організація управління ризиками;
- оцінювання ефективності та обґрунтування оптимального рівня витрат на управління ризиками.

У межах теорії та практики управління проектними ризиками найважливішими є, зокрема, методи оцінювання, моніторингу та прогнозування ризиків, інформаційного забезпечення управління ризиками [1, 3, 4].

Діяльність з управління ризиками охоплює такі основні етапи: виявлення ризику, його оцінювання, вибір методу та засобів управління ризиком, запобігання, контролювання, фінансування ризику, оцінювання результатів.

Проект – єдиний процес, що складається із сукупності скоординованих та контрольованих видів діяльності з датами початку та закінчення, здійснюється для досягнення мети, яка відповідає конкретним вимогам, і містить обмеження щодо термінів, вартості та ресурсів [6].

Проект функціонує у визначеному оточенні, що містить внутрішні і зовнішні компоненти, які, своєю чергою, враховують економічні, політичні, соціальні, технологічні, нормативні, культурні та інші фактори.

Проект завжди націлений на результат, на досягнення визначених цілей, на визначену предметну область. Реалізація проекту здійснюється повноважним керівництвом проекту, менеджером проекту і командою проекту, що працює під цим керівництвом, іншими учасниками проекту, які виконують окремі специфічні види діяльності та процеси у проекті. У роботах з проекту, як правило, на умовах часткової зайнятості, можуть брати участь представники лінійних і функціональних підрозділів компаній, відповідальних за виконання покладених на них завдань, видів діяльності, функцій, зокрема планування, керівництва, контролю, організації, адміністрування й інших загальносистемних функцій [1].

Керованими параметрами проекту є:

- обсяги робіт і види робіт проекту;
- вартість, витрати на проект;
- тимчасові параметри, що охоплюють терміни, тривалості та резерви виконання робіт, етапів, фаз проекту, а також параметри взаємозв'язку робіт;
- ресурси, необхідні для здійснення проекту, зокрема: людські чи трудові, фінансові, матеріально-технічні, поділювані на будівельні матеріали, машини, устаткування, що комплектують вироби і деталі, а також обмеження за ресурсами;
- якість проектних рішень, необхідних ресурсів, компонентів проекту тощо.

Управління проектом – це діяльність, спрямована на реалізацію проекту з максимально можливою ефективністю за заданих обмежень за часом, ресурсами, а також якістю кінцевих результатів проекту (документованих, наприклад, у технічному завданні) [6]. Для того щоб ефективно управляти обмеженнями проекту, використовують методи побудови і контролю календарних графіків робіт. Для управління ресурсними обмеженнями застосовують методи формування фінансового плану (бюджету) проекту і, у міру виконання робіт, здійснюється моніторинг, для того щоб не дати витратам вийти з під контролю. Для виконання робіт потрібне відповідне ресурсне забезпечення. Для цього використовують спеціальні методи управління людськими і матеріальними ресурсами (наприклад, матриця відповідальності, діаграми завантаження ресурсів). З трьох основних обмежень найскладнішим є контроль обмежень за заданими результатами проекту. Проблема полягає в тому, що завдання часто важко і формулювати, і контролювати. Для вирішення цих проблем використовуються, зокрема, методи управління якістю робіт.

Отже, керівники проектів відповідають за три аспекти реалізації проекту: терміни, витрати і якість результату. Однак ефективне управління термінами робіт є ключем до успіху за всіма показниками. Часові обмеження проекту часто є найкритичнішими. Там, де терміни виконання проекту серйозно затягуються, ймовірними наслідками є перевитрата коштів і недостатньо висока якість робіт. Тому у більшості методів управління проектами основний акцент робиться на календарному плануванні робіт і контролі за дотриманням календарного графіка.

Основними елементами системи управління в ситуаціях невизначеності є: виявлення в альтернативах ризику та утримання його в межах прийнятного рівня; розроблення конкретних рекомендацій, орієнтованих на усунення або мінімізацію можливих негативних наслідків ризику [1, 2]. Задача управління ризиками ІТ-проектів полягає у зменшенні впливу небажаних факторів на життєвий цикл проекту для отримання результатів, найближчих до бажаних. Для складних систем, де здебільшого вирішуються завдання дослідження і прогнозування станів залежно від вибраних стратегій управління, а саме такими є системи аналізу та управління проектними ризиками, виникають труднощі у застосуванні аналітичних моделей. Вони зумовлені такими чинниками:

- великі проекти містять багато зв'язків між процесами;
- реальні проекти потрапляють під вплив випадкових зовнішніх та внутрішніх чинників, аналітичний облік яких неможливий;
- можливість зіставлення оригіналу з моделлю існує лише на початку і після застосування математичного апарату, оскільки проміжні результати можуть не мати аналогів.

У зв'язку з цим доцільним є застосування гнучкішого методу – методу імітаційного моделювання.

Модель подається у вигляді алгоритму, в якому визначаються всі найістотніші елементи, зв'язки в системі і задаються початкові значення параметрів, що відповідають початковому моменту часу.

Усі подальші зміни, що відбуваються в системі за причинно-наслідковим законом, обчислюються за допомогою засобів логічного опрацювання даних при виконанні цього алгоритму.

Такий метод не вимагає складання рівнянь і не вимагає їх розв'язання, тому широко застосовується у багатьох сферах людської діяльності без додаткових спеціальних знань.

В ході імітаційного експерименту комп'ютер імітує функціонування системи, процесу чи проекту і обчислює характеристики властивостей, що проявляються системою.

Імітаційний експеримент подібний до натурального експерименту, проте, на відміну від натурального методу, дає змогу експериментувати з системами, яких ще або вже немає, а також передбачати поведінку наявних систем в майбутньому, вивчати їх поведінку в надзвичайних ситуаціях. Він дешевший і швидший за натурні експерименти.

Імітаційне моделювання в управлінні IT-проектами може сприйматися як своєрідний "тренажер", що дає змогу керівникові будь-якого рівня прогнозувати виконання проекту за впливу різних контрольованих і неконтрольованих чинників зовнішнього і внутрішнього середовищ [3]. За допомогою простих інструментаріїв імітаційного моделювання є можливість прораховувати вірогідність того або іншого результату при обліку впливу відразу декількох чинників. Оперативність і простота методу дає змогу варіювати величезну кількість ситуацій за безлічі комбінацій початкових умов.

Постійний процес накопичення результатів моделювання призводить до утворення хоча і достатньо великого, але скінченного масиву типових управлінських ситуацій зі скінченною кількістю стандартних способів поведінки. Це дає можливість регламентувати певні кроки в управлінні проектами і підвищити оперативність і ефективність управлінської діяльності.

Імітаційне моделювання є універсальним методом, який забезпечує як точний аналіз, так і візуальне відображення альтернативних варіантів управлінської поведінки. І внаслідок своєї універсальності, можливості численних експериментів та планування різноманітних змін імітаційні моделі складних систем є найпоширенішими.

Під час експериментів на імітаційній моделі можливим є внесення таких змін:

- у структурі моделей (додати нові елементи та зв'язки, вилучити інші);
- моделей поведінки, параметрів моделей;
- параметрів та законів розподілу випадкових факторів;
- у часі значень та зовнішніх змінних.

Імітаційні моделі дають змогу дослідити загальносистемні властивості, поведінку системи в особливих ситуаціях, знайти кращі значення параметрів системи, які до початку дослідження були вільними, прогнозувати поведінку системи в часі. Алгоритмічна структура імітаційних моделей сприяє реалізації різноманітних схем ієрархічного підпорядкування та координації між елементами моделі.

Виділимо такі вісім етапів імітаційного моделювання.

1. Визначення цілей, задач і можливостей імітації.

Ціль є зовнішньою стосовно до розглянутої системи, визначається тактичними або стратегічними інтересами особи чи групи осіб, для яких провадиться дослідження. Окремий інтерес викликає ситуація, коли ОПР (адміністратор, менеджер, керівник) розв'язує деяку практичну задачу. Ціль дослідження (або кожна з набору цілей) конкретизується у виді множини завдань, розв'язання яких необхідно і достатньо для досягнення мети. Зазвичай множина завдань має ієрархічну структуру, і її зручно зображати у вигляді дерева цілей і завдань.

Іншим чинником, що зумовлює виокремлення означеної проблеми, є можливості дослідника (теоретичні, фінансові, часові та інші). Теоретичні можливості – ті дані і методи дослідження, які дослідник може застосувати для вирішення проблеми, ґрунтуючись на своїх професійних знаннях і досвіді. Фінансові можливості обмежують використання інструментів, обсяг експериментів і спостережень, кількість спеціалістів і допоміжного персоналу. Будь-яке дослідження повинне бути виконане в певні терміни, від яких істотно залежать масштаб постановки проблеми і глибина її дослідження.

Загалом мету дослідження можна декомпозиювати так:

— опис функціонування системи (опис системи у межах уже сформульованої проблеми, тобто деякого аспекту функціонування системи);

- прогноз функціонування за різних впливів (передбачення стану системи в майбутньому за різних варіантів внутрішніх і зовнішніх впливів (сценаріїв) на систему);
- пошук найкращого варіанта функціонування системи.

2. Аналіз системи і побудова її концептуальної моделі.

Формулювання проблеми дослідження дають змогу окреслити межі об'єкта, вирізнивши його цим із навколишнього середовища. Результатом системного підходу до проблеми є концептуальна модель, яка відображає концепцію дослідження і визначається його цілями і можливостями.

Концептуальна модель містить:

- опис границь розглянутої системи;
- набір елементів системи;
- множину показників стану для кожного елемента;
- набір зв'язків між елементами системи (за необхідності вказується інтенсивність ресурсних та інформаційних потоків);
- перелік процесів, що відбуваються у системі;
- список внутрішніх і зовнішніх впливів на систему.

Найпоширенішими є вербальний (словесний) опис концептуальної моделі; також використовуються різні схеми, графіки, таблиці та діаграми, що дають змогу у наочній і стислій формі подати необхідну інформацію. Концептуальна модель може містити формалізовані елементи (блок-схеми, потокові моделі, математичні співвідношення тощо), що полегшує наступні етапи формалізації.

3. Складання і структуризація імітаційної моделі.

Наступним етапом імітаційного моделювання є формалізація отриманої концептуальної моделі, тобто побудова формальної моделі. Загальний вигляд формальної імітаційної моделі можна задати співвідношеннями:

$$x(t + \Delta t) = x(t) + (\Delta t)f_1(x(t), u(t), \xi(t)),$$

$$x(0) = x_0, t = 0, \dots, T - \Delta t,$$

де

$x(t) = (x_1(t), \dots, x_n(t))$ – вектор стану;

$x_i(t), (i = 1, \dots, n)$ – це значення показників, що характеризують стан системи у момент часу t ; компоненти вектора стану можуть набувати як кількісних (числових, символічних тощо), так і якісних значень на деякій порядковій шкалі;

$u(t) = (u_1(t), \dots, u_m(t))$ – вектор керованих впливів;

$\xi(t) = (\xi_1(t), \dots, \xi_p(t))$ – вектор некерованих впливів;

$f_1(x(t), u(t), \xi(t))$ – вектор-функція тієї самої розмірності, що і $x(t)$, яка визначає динаміку вектора стану в умовах зовнішнього впливу;

$x_0 = (x_{01}, \dots, x_{0n})$ – початкове значення вектора стану, яке вважається відомим;

T – період, протягом якого моделюється динаміка системи;

Δt – крок моделювання (різниця в часі між двома послідовними станами системи).

Кожний з векторів $x(t)$, $u(t)$ і $\xi(t)$ може набувати значення з якоїсь допустимої області, що позначається відповідно $X(t)$, $U(t)$ і $\Xi(t)$. Область допустимих станів $X(t)$ – характеризує діапазон можливих станів системи, область допустимих керувань $U(t)$ – характеризує можливості суб'єкта з керування системою, а область $\Xi(t)$ – характеризує можливі значення неконтрольованих факторів (за цілковитої відсутності інформації ця область збігається з усім простором R^p).

Залежно від мети моделювання вводиться системний час, що моделює хід часу у реальній системі. Розрізняють два типи шкал модельного (системного) часу: рівномірний і подійний (за діями). Після задавання вектора стану системи і вибору часового кроку розробляється декомпозиція моделі і визначається її блокова конструкція. Блоковий принцип побудови моделі має цілу низку переваг, особливо відчутних під час створення складних імітаційних моделей.

4. Програмна реалізація імітаційної моделі.

Невід'ємним етапом прикладного системного дослідження є виконання комп'ютерних імітаційних експериментів з моделлю. Побудована формальна модель повинна бути представлена у вигляді програми, яка реалізує відповідний алгоритм дослідження: числове розв'язання системи алгебраїчних і диференціальних рівнянь, задача оптимізації, перехід системи з початкового стану у кінцевий.

5. Аналіз і корекція імітаційної моделі.

Перед використанням побудованої імітаційної моделі необхідно вирішити такі питання:

- вибрати значення структурних і числових параметрів моделі;
- впевнитися в тому, що за цих значень параметрів модель добре відповідає моделюваній системі, адекватно її описує (верифікація, перевірка певної адекватності).

6. Планування і виконання імітаційних експериментів.

В імітаційному моделюванні комп'ютер відіграє роль експериментної установи, видає значення керованих змінних, неконтрольованих факторів і початкових умов траєкторії системи. Оскільки кількість можливих поєднань впливу керованих та некерованих зовнішніх чинників (навіть для скінченної їх множини) на систему велика, то є зрозумілою роль планування імітаційних експериментів.

7. Опрацювання і аналіз результатів імітації.

Зазвичай безпосередні результати імітації ще не придатні для розв'язання поставлених задач. Їх треба систематизувати, подати у вигіднішому для подальшого аналізу вигляді, надати ОПР та проаналізувати. На цій стадії можна отримати негативні результати, що означатиме необхідність корекції моделі і повернення до попередніх етапів.

До статистичних методів опрацювання результатів моделювання належать:

- фіксування даних і накопичення статистики моделювання;
- визначення інтервалів для вихідних величин;
- виявлення функціонального зв'язку між змінними за допомогою регресійного аналізу;
- ідентифікація закону розподілу по гістограмі.

Аналіз результатів моделювання містить:

- оцінку точності імітаційного експерименту;
- зменшення кількості параметрів моделі;
- визначення інтервалів зміни параметрів;
- пошук джерела помилок;
- видалення різко відхилених значень;
- вибір системи координат для подання результатів тощо.

8. Ведення і супровід результатів імітації.

Головним у імітаційному моделюванні систем є розв'язання практичних задач. Тому процес імітаційного моделювання не закінчується навіть після отримання, опрацювання та інтерпретації результатів імітації ОПР та експертами. Потрібно забезпечити практичне використання цих результатів в управлінні відповідної динамічної системи. Потрібно, щоб результати могли бути використані не тільки для єдиного розв'язання поставленої задачі, але і для багатократного вирішення комплексу проблем, пов'язаних з цією системою [3].

Отже, імітаційні моделі дають змогу дослідити поведінку великих систем, зокрема IT-проектів, які не можна досліджувати за допомогою інших методів.

Цілі статті

Дослідження імітаційних моделей та їх застосування для управління ризиками інформаційно-технологічних проектів.

Основний матеріал

Подамо інформаційно-технологічний проект як певну послідовність процесів $w_i \in W$. Процес – це сукупність взаємопов'язаних або взаємодійних видів діяльності, яка перетворює входи на виходи [6]. Кожен процес має свої визначені терміни початку $t_i^N \in T$ та завершення $t_i^3 \in T$. Причому початок деяких процесів може залежати від завершення інших, що йому передують.

Модель ІТ-проекту будемо подавати у вигляді мережі. Кожен процес проекту позначатиметься дугою мережі, яка орієнтована за напрямом виконання проекту і поєднує два етапи проекту. Етапи проекту позначаються вузлами мережі і встановлюють стосунки передування серед процесів ІТ-проекту. Інакше кажучи, етап – це подія початку чи завершення процесу.

Побудова мережі проекту ґрунтується на таких правилах:

1. Кожен процес подається однією і лише однією дугою.
2. Кожен процес ідентифікується двома кінцевими вузлами (вузол початку процесу і вузол його завершення).
3. Для підтримання правильних стосунків передування у разі введення в мережу будь-якого додаткового процесу необхідно вказати:
 - а) який процес безпосередньо передує доданому процесу;
 - б) який процес повинен виконуватися після завершення доданого процесу;
 - в) який процес конкурує, тобто виконується паралельно з доданим процесом.

На рис.1 наведено приклад мережевої моделі проекту, в якому є десять подій (вузли 1, 2, ..., 10), що визначають завершені етапи проекту. Напрявлені дуги між вузлами мережі визначають процеси проекту (14 процесів), які характеризуються певною тривалістю виконання та певними ресурсними обмеженнями.

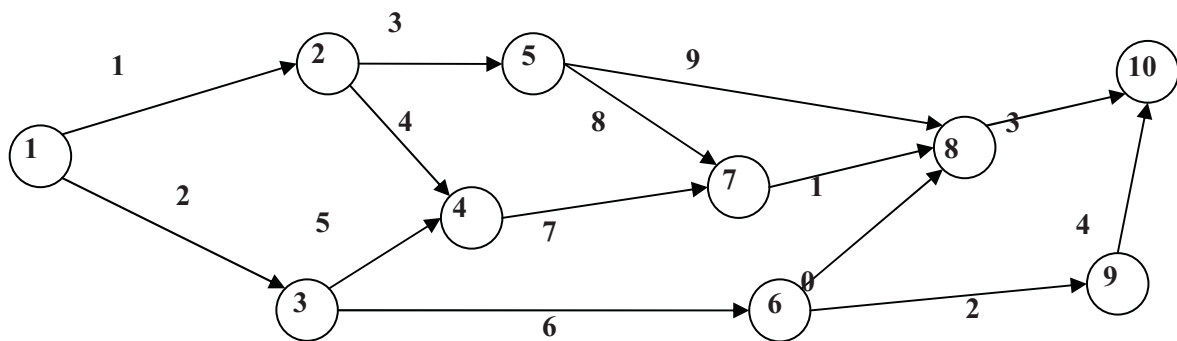


Рис. 1. Приклад мережевої моделі ІТ-проекту

Аналіз ризику дотримання термінів виконання проекту полягає у визначенні імовірності, з якою проект буде виконаний у визначений час і навпаки, скільки часу потрібно для того, щоб із заданою імовірністю успішно завершити проект. Очевидно, що проект буде успішно, тобто у визначений наперед термін, завершений у випадку вчасного завершення усіх робіт проекту. Тривалість ходу виконання проекту може бути обчислена за допомогою підсумовування найдовшого за хронологічним виміром шляху (критичного шляху), який визначає максимальну тривалість виконання проекту загалом.

Процес є критичним, якщо він не має “припуску” для часу свого початку і завершення. Отже, щоб весь проект завершився без затримок, необхідно щоб всі критичні процеси починалися і закінчувалися у строго визначений час. Для некритичного процесу можливі деякі відхилення часу його початку, але в певних межах, тобто коли час його початку не впливає на тривалість виконання всього проекту.

Для виконання необхідних обчислень визначимо подію як точку на часовій осі, де завершується один процес і починається інший. Введемо такі позначення:

- t'_j – найраніший можливий час настання події j ;
- t''_j – найпізніший можливий час настання події j ;
- D_{jq} – тривалість процесу (j, q) .

Обчислення критичного шляху відбувається за два проходи. За перший прохід обчислюються найраніші часи настання подій, а за другий – найпізніші часи настання тих самих подій.

I прохід

Обчислення починаються у вузлі 1 і закінчуються в останньому вузлі n .

Вважаємо, що $t'_j = 0$, тобто проект починається в нульовий момент часу. Далі для кожного вузла j визначаються вузли p, q, \dots, v , безпосередньо зв'язані з вузлом j процесами (p, j) (q, j) , ..., (v, j) , для яких вже обчислені найраніші часи настання відповідних подій. Найраніший час настання події j обчислюється за формулою: $t'_j = \max \{t'_p + D_{pj}, t'_q + D_{qj}, t'_v + D_{vj}\}$.

Прохід вперед завершується, коли буде обчислена величина t'_n для вузла n .

II прохід

Обчислення починаються в останньому вузлі n і закінчуються у вузлі 1.

Вважаємо, що $t''_n = t'_n$, тобто найраніший і найпізніший часи для завершення проекту збігаються. Далі для кожного вузла j визначаються вузли p, q, \dots, v , безпосередньо зв'язані з вузлом j процесами (p, j) (q, j) , ..., (v, j) , для яких вже обчислені найпізніші часи настання відповідних подій. Найпізніший час настання події j обчислюється за формулою: $t''_j = \min \{t''_p - D_{pj}, t''_q - D_{qj}, t''_v - D_{vj}\}$.

Прохід назад завершується, коли буде обчислена величина t''_1 для вузла 1.

Процес (i, j) буде критичним, якщо виконуються такі умови:

1. $t''_i = t'_i$;
2. $t''_j = t'_j$;
3. $t''_j - t''_i = t'_j - t'_i = D_{ij}$.

Критичні процеси повинні утворювати неперервний шлях через усю мережу від початкової події до кінцевої.

Для реалізації імітаційної моделі аналізу ризиків ІТ-проектів розроблено систему, яка розв'язує завдання визначення імовірності успішного завершення проекту у задані терміни та час завершення проекту за заданою імовірністю успішного завершення.

Проект задається у вигляді мережі. Кожний процес проекту позначається дугою, орієнтованою за напрямом виконання проекту. Кожна подія задається вузлом мережі і встановлює стосунки передування серед процесів проекту. Кожний процес проекту характеризується своїм номером, назвою роботи та тривалістю.

Тривалості процесів можуть бути детерміновані, стохастичні або вірогіднісні. У першому випадку використовують метод CPM (Critical Path Method), у другому – метод PERT (Program Evaluation and Review Technique).

У разі вірогіднісних тривалостей процесів для кожного з них необхідно отримати вірогіднісні розподіли часу виконання цього процесу. Для отримання закону вірогіднісного розподілу та його параметрів можна скористатись декількома способами. Перший – отримання виду та параметрів закону розподілу за допомогою опитування експертів (суб'єктивні вірогідності). Другий ґрунтується на множині проектів-аналогів (об'єктивні вірогідності). Для визначення проектів-аналогів необхідно спершу класифікувати модельований проект. Для цього використовуємо базу вже виконаних проектів і за допомогою нечіткої класифікації розділимо проекти на класи за певними ознаками. Опісля, виконавши кластерний аналіз модельованого проекту, визначаємо клас, до якого він належить, та вибираємо процедуру нормування розрахунку параметрів проекту.

Для прикладу роботи розробленої системи візьмемо проект, заданий мережею на рис. 1. Вхідними даними у цій системі для введення мережевої моделі проекту є характеристики процесів проекту: номери процесів та їхні назви, найдовша та найкоротша тривалості процесу. Для встановлення стосунків передування серед процесів проекту як вхідний параметр для кожного процесу задають номери процесів, без виконання яких неможливий початок цього процесу. Ці дані заносять за допомогою діалогового вікна (рис. 2) у таблицю (рис. 3).

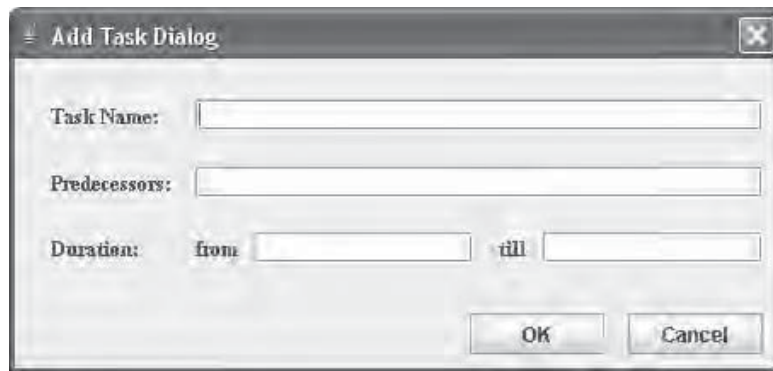


Рис. 2. Діалогове вікно введення вхідних даних

#	Task Name	Predecessors	From Duration	Till Duration
1	name1		12	15
2	name2		21	23
3	name3	1	7	11
4	name4	1	14	16
5	name5	2	17	20
6	name6	2	11	15
7	name7	4,5	8	11
8	name8	3	7	9
9	name9	3	3	6
10	name10	6	6	9
11	name11	7,8	9	11
12	name12	6	12	14
13	name13	9,11,10	9	11
14	name14	12	23	26

Рис. 3. Таблиця із введеними вхідними даними тестового проекту

Для всіх процесів проекту вказується закон вірогіднісного розподілу, який буде використовуватись генератором випадкових величин, та за необхідності потрібні параметри розподілу. Окремо задається кількість розрахункових ітерацій (рис. 4). Для надання випадкових значень за вибраним законом розподілу використовують генератори випадкових величин.

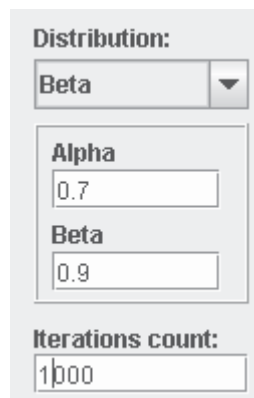


Рис. 4. Вікно введення виду та параметрів розподілу

Зокрема, у системі передбачені генератори випадкових величин для надання випадкових значень за такими законами розподілу:

- рівномірний розподіл;
- нормальний розподіл (параметр D);
- β -розподіл (параметри α і β);
- експоненціальний розподіл;
- γ -розподіл (параметр α).

Система працює так. Після задоволення вхідних даних відбувається генерація випадкових тривалостей робіт з проміжку, заданого користувачем. Генерація відбувається за вибраним законом розподілу. Опісля знаходять критичний шлях проекту, відповідно до заданого порядку слідування робіт та їх згенерованих тривалостей. Кількість таких переходів залежить від кількості введених розрахункових ітерацій.

Для проекту, заданого таблицею (рис. 3) і параметрами розподілу (рис. 4), критичні шляхи будуть такими, як зображено на рис. 5.

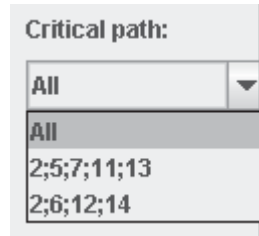


Рис. 5. Критичні шляхи для тестового проекту

Далі система формує вірогідні розподіли для знайдених критичних шляхів, аналізує їх та зображає графічно (рис. 6).

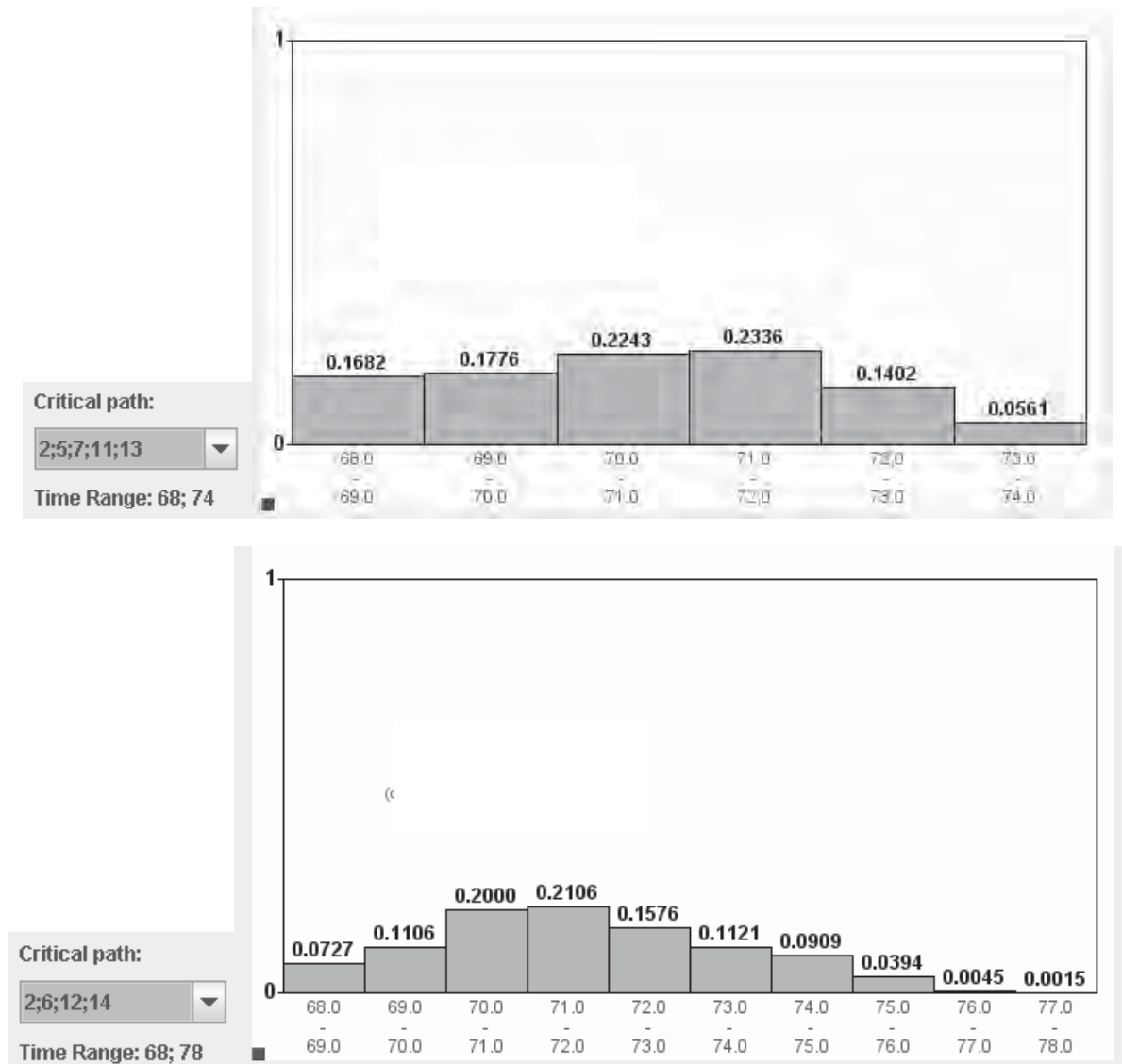


Рис. 6. Ймовірнісні розподіли за критичним шляхами

Вихідними даними системи є сформовані вірогіднісні розподіли як по окремих знайдених критичних шляхах, так і для проекту загалом (рис. 7).

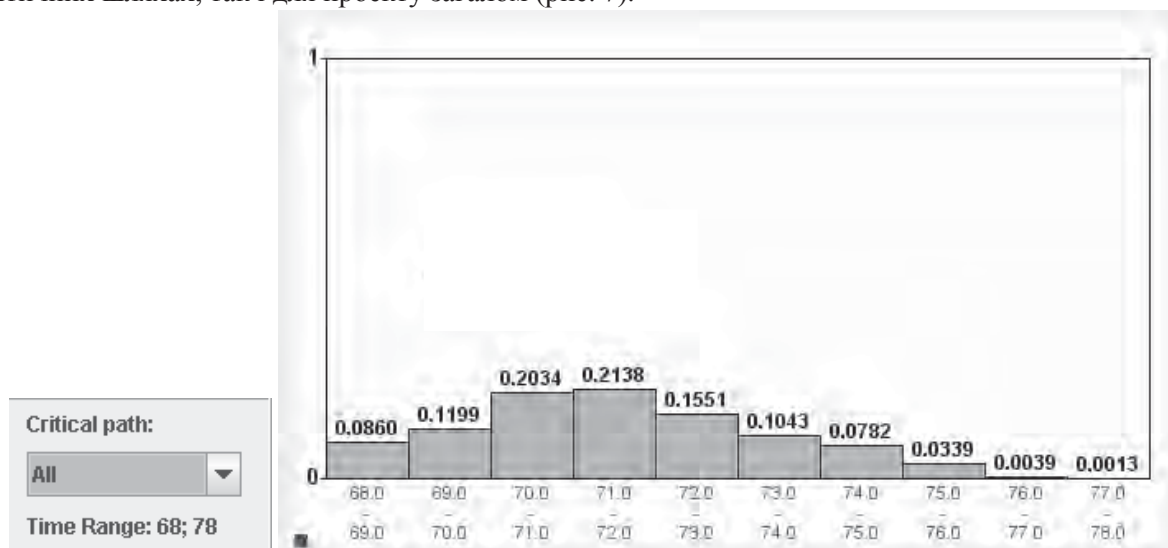


Рис. 7. Ймовірнісний розподіл по усьому проекту

У результаті роботи системи ми можемо отримати відповідь на запитання:

- за який час можна виконати проект із наперед заданою вірогідністю?
- з якою вірогідністю може бути виконаний проект у заданий проміжок часу?

Наприклад: за термін 72 часові одиниці проект, який заданий мережею на рис. 1, буде завершений з вірогідністю 0.80373, а завершення цього самого проекту з вірогідністю 0.9 можливе у термін, не більший за 73 часові одиниці (рис. 8).

Enter Time:

Probability

Enter Probability:

Time

Рис. 8. Термін та вірогідність успішного завершення проекту

Отримана інформація може бути використана для вироблення конкретних рекомендацій щодо зменшення впливу небажаних факторів на життєвий цикл ІТ-проекту з метою одержання результатів, найближчих до бажаних.

Висновки

ІТ-проекти характеризуються великою кількістю взаємопов'язаних процесів. Найдоцільнішим методом для аналізу та управління ризиками ІТ-проектів, зокрема ризиків дотримання календарних графіків, є імітаційне моделювання. Цей метод забезпечує як точний аналіз, так і візуальне представлення альтернативних варіантів управлінської поведінки. Подана імітаційна модель дає змогу побудувати розподіли значень вірогідностей для найважливіших параметрів перебігу реалізації проекту, що, своєю чергою, дає змогу висувати та перевіряти стохастичні гіпотези. Подальша робота у цьому напрямі буде зосереджена на дослідженнях ситуацій, у яких кожний конкретний процес ІТ-проекту може мати інший вид вірогіднісного розподілу часу виконання.

1. Управління ризиками в проектній діяльності / О.М. Верес, А.В. Катренко, І.В. Рішняк, В.М. Чаплига // Інформаційні системи та мережі. Вісник Національного університету "Львівська політехніка". – 2003. – № 489. – С.38–49. 2. Катренко А.В. Методи управління ризиками в ІТ-проектах / А.В. Катренко, І.В. Рішняк // Комп'ютерні науки та інформаційні технології (CSIT-2008): III міжнародна наук.-практ. конф., 25–27 вересня 2008 р.: тези доповіді – Львів, 2008. – С.245–247. 3. Катренко А.В. Імітаційне моделювання в управлінні проектними ризиками / А.В. Катренко, І.В. Рішняк // Сучасні засоби та технології розроблення інформаційних систем: Міжнародна науково-практична конференція, ХНЕУ, 20–21 листопада 2008 р.: тези доповіді – Харків, 2008. – С.75–76. 4. Модель управління проектними ризиками / І.В. Рішняк // Комп'ютерні системи проектування. Теорія і практика. Вісник Нац. ун-ту "Львівська політехніка". – 2004. – № 522. – С.155–160. 5. Рішняк І.В. Структурна модель, основні складові та класифікація проектних ризиків / І.В. Рішняк // Інтелектуальні системи прийняття рішень та прикладні аспекти інформаційних технологій (ISDMIT-2007): наук.-практ. конф., 14–18 травня 2007 р.: тези доповіді – Євпаторія, 2007. – Т. 2. – С.214–217. 6. DСТУ ISO 9000-2001. 7. Chapman C., Ward S., Project risk management: processes, techniques and insights. Chichester: John Wiley & Sons, 1997. 8. Kahkonen K., Integration of risk and opportunity thinking in projects, Fourth European Project Management Conference, PMI Europe 2001, London UK, 6–7 June 2001.

УДК 004.9

Т.М. Рудакевич, Г.Г. Цегелик

Львівський національний університет імені Івана Франка,
кафедра математичного моделювання соціально-економічних процесів

РОЗШИРЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ СКБД POSTGRESQL ДЛЯ ОПТИМІЗАЦІЇ ПОШУКУ ІНФОРМАЦІЇ У ФАЙЛАХ БАЗ ДАНИХ

© Рудакевич Т.М., Цегелик Г.Г., 2010

Розглянута задача реалізації методу пошуку інформації у файлах БД, який враховує розподіл імовірностей звертання до записів, в СКБД PostgreSQL через розширення її функціональних можливостей. Наведено розв'язок підзадачі – реалізацію аналізатора статистики звертань до записів.

Ключові слова: БД, алгоритми пошуку, розширення функцій СКБД.

The PostgreSQL DBMS was extended by statistics analyzer realization. It is a part of search method that is based on allocation of entry's selection probabilities.

Keywords: Database Management System, search algorithm, probability theory.

Вступ

У світі математики та інформатики завжди існує тенденція до прогресу та руху вперед, відкриття якихось нових закономірностей, оптимізації алгоритмів та впровадження їх у життя.

Оскільки основу сучасних інформаційних технологій становлять бази даних (БД) і системи керування базами даних (СКБД), то удосконалення технології опрацювання інформації з використанням концепції БД передбачає, передусім, вирішення проблеми оптимальної організації та пошуку інформації у файлах баз даних, що, своєю чергою, забезпечує доступ користувачів до інформації БД за мінімально допустимий час.

Враховуючи те, що в багатьох системах опрацювання інформації типовими є випадки нерівномірного розподілу ймовірностей звертання до записів файлів, то актуальною сьогодні є