Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 2 (8)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

УДК 004.4'22 + 658.5
DOI: https://doi.org/10.30837/2522-9818.2019.8.025

С. С. Великодний, Ж. В. Бурлаченко, С. С. Зайцева-Великодна

РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМНОГО ЗАСОБУ ДЛЯ УПРАВЛІННЯ МЕРЕЖЕВИМ ПЛАНУВАННЯМ РЕІНЖИНІРИНГУ ПРОГРАМНОГО ПРОЕКТУ

Предмет дослідження – програмний засіб побудови графічної мережевої моделі реінжинірингу програмного проекту. **Мета** дослідження – розробка системної архітектури програмного засобу для автоматизованого проектування мережевих графіків організації виробництва з реінжинірингу програмних систем у рамках управління проектами. Поряд з лінійними графіками й табличними розрахунками мережеві методи планування знаходять широке застосування при розробці перспективних планів і моделей створення складних виробничих систем та інших об'єктів довгострокового використання. **Завданням** перед створенням програмного засобу є здатність працювати з усіма типами мережевих графіків із можливостями їх всебічної трансформації. **Методи.** В основу статті закладено методи мережевого планування за методологію PERT (Program (Project) Evaluation and Review Technique), використання елементів теорії графів та методу діаграм Ганта, як облікового для здійснення управління проектами. Моделювання системної архітектури програмного забезпечення виконується у рамках методології UML (Unified Modeling Language) 2.5 із використанням CASE-інструментарію Enterprise Architect 14. **Результати.** У статті розроблено програмний засіб для управління мережевим плануванням реінжинірингу програмного проекту. **Висновки.** Архітектура розроблена у вигляді декількох структурних та поведінкових діаграм, а саме: діаграма варіантів використання, що надає аналітику детальної уяви про галузь застосування програмного засобу; діаграма послідовності, яка призначена для формування уяви програміста про порядок виконання дій при роботі з майбутнім програмним засобом; діаграма станів, що необхідна для наочного подання тих станів, у яких програмний засіб може знаходитися у різні моменти часу; діаграма класів, яка використовуються для проектування основного формового наповнення майбутнього програмного засобу; діаграма компонентів, що призначена для вивчення складу компонентів майбутнього програмного засобу та вказівки послідовності компіляції та збірки окремих модулів. Чисельна та часова оцінка параметрів планування будується за даними, що отримані із проектних діаграм Ганта.

Ключові слова: управління проектом; граф; мережевий графік; програмний засіб; реінжиніринг; CASE-засіб; UML-діаграма.

Вступ

Процес створення проекту, прототипу, прообразу майбутнього об'єкта, стану та способів його виготовлення називається проектуванням. У техніці під проектуванням розуміють розробку проектної, конструкторської та іншої технічної документації, призначеної для забезпечення створення нових видів та зразків. У проектуванні застосовують системний підхід, який полягає у встановлені структури системи, типу зв’язків, визначені атрибутів, аналізуванні впливів зовнішнього середовища. В процесі проектування виконуються технічні та економічні розрахунки, схеми, графіки, пояснювальні записки, кошториси, калькуляції та описи.

Однією з головних складових проектування – є планування, що визначається як заздалегідь намічений порядок дій або оптимальний розподіл ресурсів, необхідних для досягнення поставленої мети.

Мережеве планування – це одна з форм графічного відображення змісту робіт й тривалості виконання стратегічних планів і довгострокових комплексів проектних, планових, організаційних та інших видів діяльності підприємства.

Поряд з лінійними графіками й табличними розрахунками мережеві методи планування знаходять широке застосування при розробці перспективних планів і моделей створення складних виробничих систем та інших об'єктів довгострокового використання. Мережеві плани робіт підприємств по створенню нової конкурентоздатної продукції містять не тільки загальну тривалість всього комплексу проектно-виробничої та фінансово-економічної діяльності, але й тривалість та послідовність здійснення окремих процесів чи етапів, а також потребу необхідних економічних ресурсів.

Завданням, що ставиться перед створенням програмного засобу управління мережевим плануванням є здатність працювати з усіма типами мережевих графіків із можливостями їх всебічної трансформації.

Аналіз досліджень і публікацій

Поняття "мережевий графік", згідно з [1] – це динамічна модель виробничого процесу, яка відображає технологічну залежність та послідовність виконання комплексу робіт, що погоджує їх завершення у часі з урахуванням витрат ресурсів й вартості робіт з виділенням при цьому вузьких (критичних) місць. Одночасно, мережевий графік – граф, вершини якого відображають стани деякого об'єкта (наприклад, будівництва), а дуги – роботи, що ведуться на цьому об'єкті. Часто мережевий графік будується так, що розташування вершин по горизонталі відповідає часу досягнення стану, відповідного заданої вершини (популярна складова методології PERT – Program (Project) Evaluation and Review Technique).

Мережевий графік заснований на використанні математичної моделі – графа. Будь-яка послідовність робіт у мережевому графіку, в якому кінцева подія кожної роботи цієї послідовності, збігається з початковою подією наступної за нею роботою, називається шляхом. Для різних областей використання види графів можуть відрізнятися орієнтованістю, обмеженнями на кількість зв'язків і додатковими даними про вершини або ребра. Велика

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2019. No. 2 (8)

кількість структур, які мають практичну цінність в математиці та інформатиці, можуть бути представлені графами [2, 3]. Наприклад, будову Вікіпедії можна змоделювати за допомогою орієнтованого графу, в якому вершини – це статті, а дуги (орієнтовані ребра) – посилання на інші статті.

Для опису графів в цілях, придатних для машинної обробки і, одночасно, зручному для людського сприйняття – використовується кілька стандартизованих мов [4], серед яких: DOT (мова), Graph ML, Trivial Graph Format, GML, GXL, XGMML, DGML, Java [5].Відзначимо спеціалізовані програми для побудови графів [6]. До найбільш вдалих відносяться комерційні ILOG, GoView, Lassalle Add Flow, LEDA. З безкоштовних можна відзначити Boost Graph Library. Для візуалізації графів можна використовувати Graphviz (на думку експертів, вона добре працює для орграфов) або LION GraphVisualizer [7]. Особливо відзначимо програму Графоаналізатор – це російськомовна програма, з вельми простим для користувача інтерфейсом [8].

Діаграми Ганта (англ.Gantt chart, стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Ці діаграми є одним з методів планування та управління проектами [9]. Діаграма Ганта представляє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями. Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту "сьогодні". Найчастіше діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці – містять додаткову інформацію про задачу [10].

PERT (англ.) – техніка оцінки та аналізу програм (проектів), яка використовується при управлінні проектами [11]. PERT – це спосіб аналізу завдань, необхідних для виконання проекту, особливо, аналізу часу, який потрібен для виконання кожної окремої задачі, а також визначення мінімального необхідного часу для виконання всього проекту (рис. 1) [12].

Найпопулярнішою частиною PERT – є метод критичного шляху, що спирається на побудову мережевого графіку (мережевої діаграми PERT). Метод критичного шляху – ефективний інструмент планування розкладу та управління термінами проекту. В основі методу лежить визначення найбільш тривалої послідовності завдань від початку проекту до його закінчення з урахуванням їх взаємозв'язку. Завдання, що лежать на критичному шляху (критичні завдання) мають нульовий резерв часу виконання та у разі зміни їх тривалості змінюються терміни всього проекту.

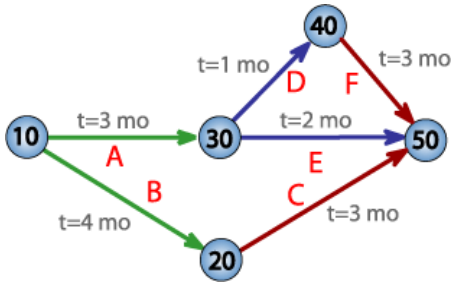

Рис. 1. Приклад мережевої діаграми PERT для проекту тривалістю у сім місяців із п'ятьма проміжними точками (від 10 до 50) і шістьма діяльностями (від A до F)

У зв'язку з цим при виконанні проекту критичні завдання вимагають більш ретельного контролю, зокрема, своєчасного виявлення проблем та ризиків, що впливають на терміни їх виконання і, отже, на строки виконання проекту в цілому [13]. У процесі виконання проекту критичний шлях проекту може змінюватися тому, що при зміні тривалості задач деякі з них можуть опинитися на критичному шляху. Найвідоміша частина PERT – це діаграми взаємозв'язків робіт і подій. PERT пропонує використовувати діаграми-графи з роботами на вузлах, з роботами на стрілках (мережеві графіки), а також діаграми Ганта.

Виділення невирішених раніше частин загальної проблеми

На вітчизняних підприємствах, циклові або лінійні графіки, звичайно, застосовуються у процесі короткострокового чи оперативного планування виробничої діяльності. Основним недоліком таких планів-графіків – є відсутність можливості тісної взаємозв'язки окремих робіт в єдину виробничу систему або загальний процес досягнення запланованих кінцевих цілей підприємства (фірми). Мережеві графіки служать не тільки для планування різноманітних довгострокових робіт, але і для їх координації між керівниками та виконавцями проектів [14], також мережеві графіки необхідні для визначення необхідних виробничих ресурсів та їх раціонального використання.

Автоматизовані системи планування ресурсів виробництва (ERP), зазвичай включають комп'ютерні програми [15], які у тій, чи іншій мірі автоматизують деякі етапи складання та коригування мережевих графіків, проте подібні програми мають досить високу ліцензійну ціну та непід'ємні для вітчизняного виробника.

Таким чином, **метою дослідження** є – розробка системної архітектури програмного засобу (ПЗ) для автоматизованого проектування мережевих графіків організації виробництва з реінжинірингу програмних систем у рамках управління проектами.

Об’єктом роботи є мережеве планування виробничого процесу.

Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 2 (8)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

Предметом дослідження є ПЗ побудови графічної мережевої моделі реінжинірингу програмного проекту.

Матеріали і методи

В основу статті закладено: мережеве планування за методологію PERT, використання елементів теорії графів та методу діаграм Ганта, як облікового для здійснення управління проектами. Моделювання системної архітектури програмного забезпечення виконується у рамках методології UML (Unified Modeling Language) 2.5 із використанням CASE-інструментарію Enterprise Architect 14.

Результати досліджень

Розробку системної архітектури програмного засобу для управління мережевим плануванням реінжинірингу програмного проекту (далі – ПЗ) розпочнемо із проектування діаграми варіантів використання. Вони використовуються для надання аналітику детальної уяви про галузь застосування програмного забезпечення, що розробляється. З діаграми варіантів використання стає зрозуміло для чого призначений продукт, які підсистеми та модулі він має, якими зв'язками поєднані елементи та сутності у ньому. Центральним елементом спроектованої діаграми варіантів використання – є мережевий графік (МГ), який зображено у вигляді актору зі стереотипом "business actor" (рис. 2). Від актору "МГ" відходять різноманітні зв'язки, більшість з яких асоціації зі стереотипом "uses" (використання), проте також присутні залежності. З діаграми варіантів використання видно, що МГ використовуються у всіх галузях застосування, пов'язаних із наведеним переліком.

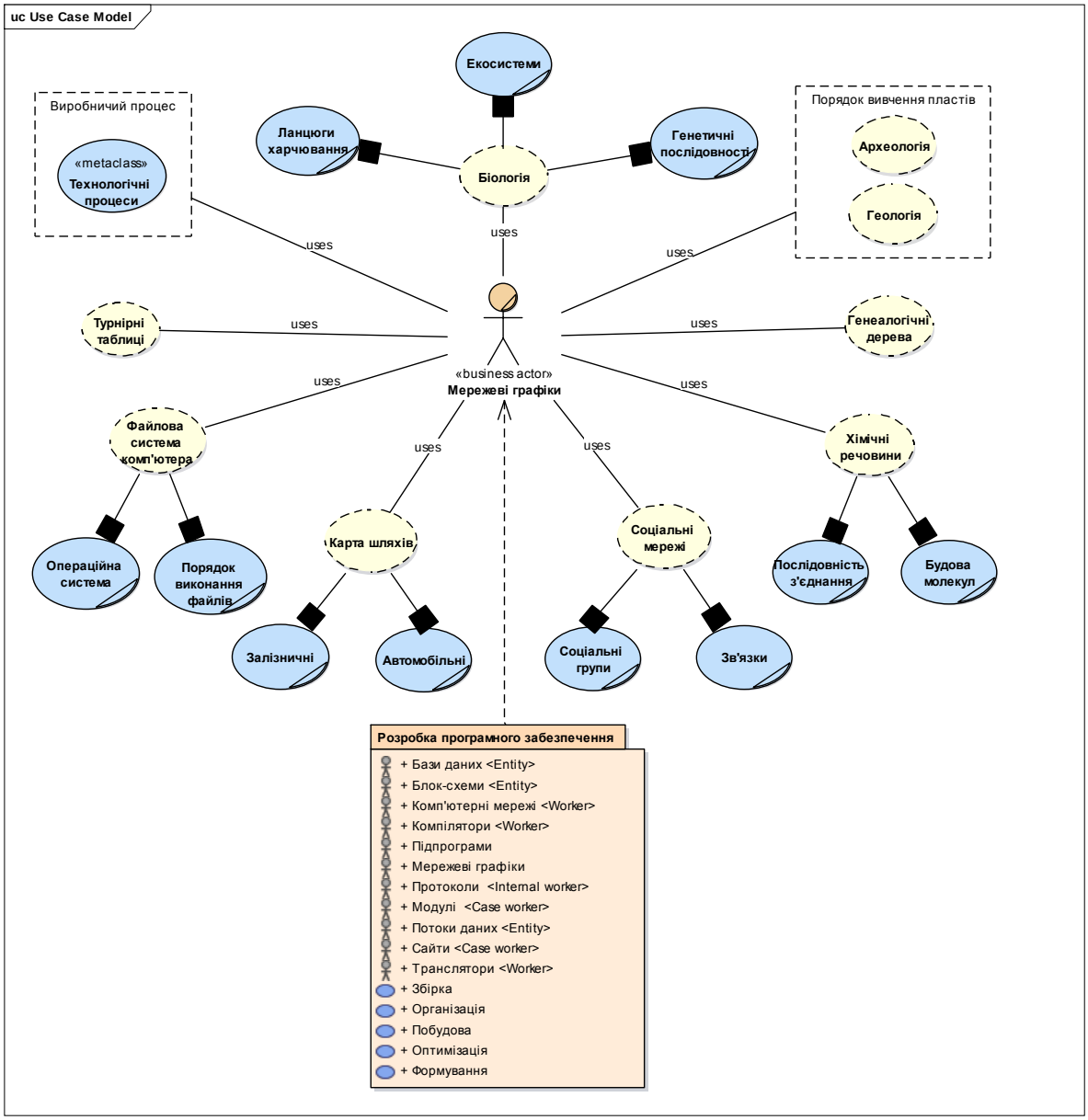

Рис. 2. Діаграма варіантів використання

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2019. No. 2 (8)

Перейдемо до розгляду сутності, що об’єднана з МГ типом зв’язку залежність "dependency". На діаграмі варіантів використання – це пакет "Розробка програмного забезпечення", виконання якого дійсно базується (залежить) від МГ. До складу пакету входять багато сутностей, об’єднаних зв’язками, сукупність яких являє вкладену спроектовану піддіаграму діаграми варіантів використання (рис. 3), що має графоподібну топологію.

Інженерія програмного забезпечення та комп'ютерні науки взагалі є однією з тих галузей, де МГ застосовуються найчастіше, тому поданий пакет й винесено у вигляді окремої залежної структури (рис. 3). Складність та велика кількість модулів і протоколів у сучасних програмних продуктах сильно ускладнює розуміння їх роботи, керування нею та її оптимізацію. Тому дуже часто складаються МГ програм, причому найчастіше це робиться автоматично трансляторами, компіляторами чи парсерами.

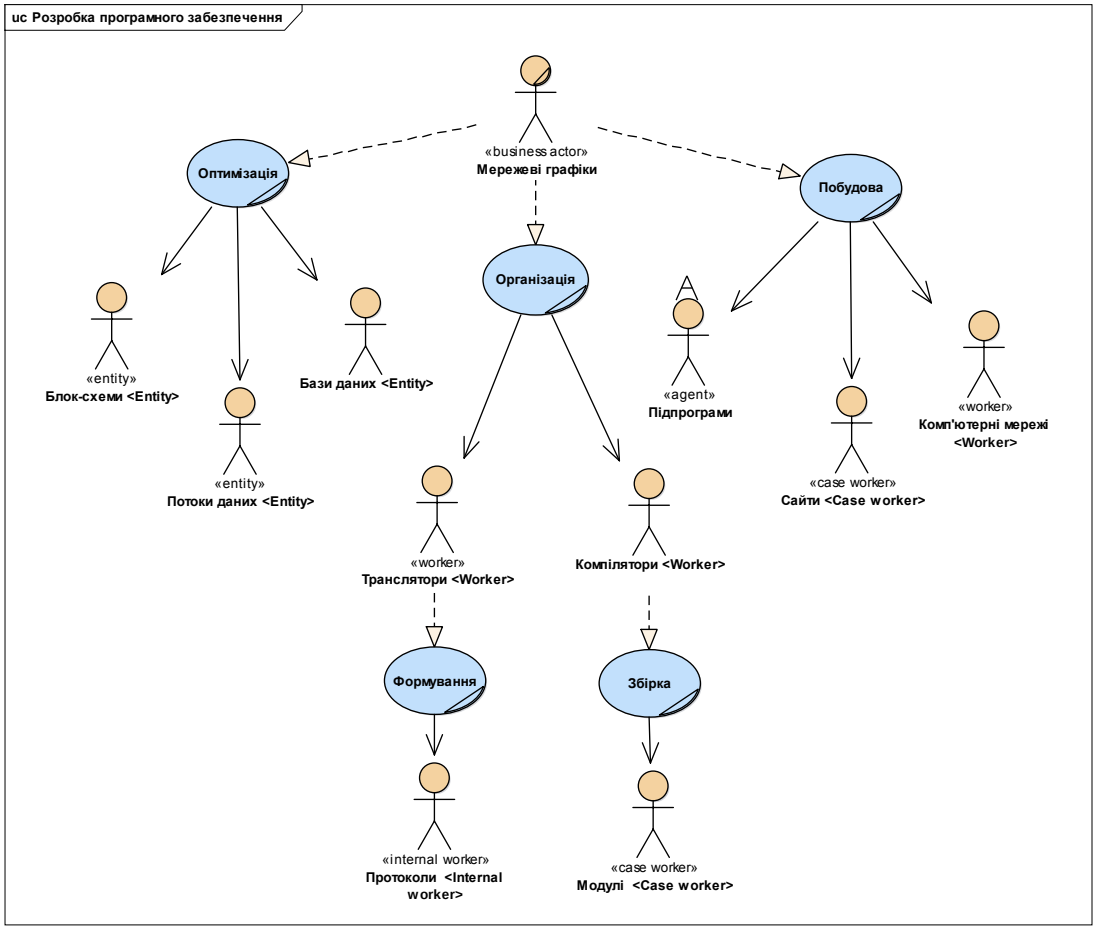

Рис. 3. Діаграма-пакет "Розробка програмного забезпечення"

Діаграми послідовності призначені для формування уяви програміста про порядок виконання дій при роботі з майбутнім ПЗ, які сформовані системним архітектором. Стандартно існує діаграми послідовності двох основних типів: для відображення дій програміста при розробці ПЗ та для відображення дій користувача при подальшій роботі з майбутнім ПЗ. Діаграми послідовності саме другого типу використовуються для створення інструкцій користувача, які є складовою частиною програмно-методичних комплексів (ПМК).

Саме такий тип діаграм послідовності спроектовано у поданій статті (рис. 4), оскільки для виконання реінжинірингу програмного проекту, необхідно розуміти принципи роботи ПЗ.

Діаграма станів необхідна для наочного подання тих станів ПЗ, у яких він може знаходитися в різні моменти часу. Іншими словами, можна сказати, що стани описують поведінку ПЗ, яка, в свою чергу, може залежати як від вказівок користувача, так і від обчислювальних процедур самого ПЗ.

Як видно з самої назви діаграми – ключовими моментами у ній – є стани (state), які представляють собою, у своєму роді, точки зупинки роботи ПЗ або збору статистичної та технічної інформації (протоколювання). Кожен стан має точки входу та виходу у цей стан, причому, таких точок може бути декілька (як для входів так і для виходів) – саме для фіксування цих позицій у межах стану – існує таке поняття, як «історія стану», яка опційно вмикається у специфікації кожного стану та має позначення: літера "H" (history), що подається у колі.

Крім того, важливими на діаграмах станів, також, є й переходи (transfer), що являють собою зв’язки між

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 2 (8) ISSN 2524-2296 (online)

станами. Кожен перехід має синтаксис, який розкриває математичний або фізичний процес, що викликав цей перехід. Синтаксис, у свою чергу, крім назви переходу, може містити обмежуючу умову (подається у прямокутних дужках [ ]) – при невиконанні якої – перехід неможливий.

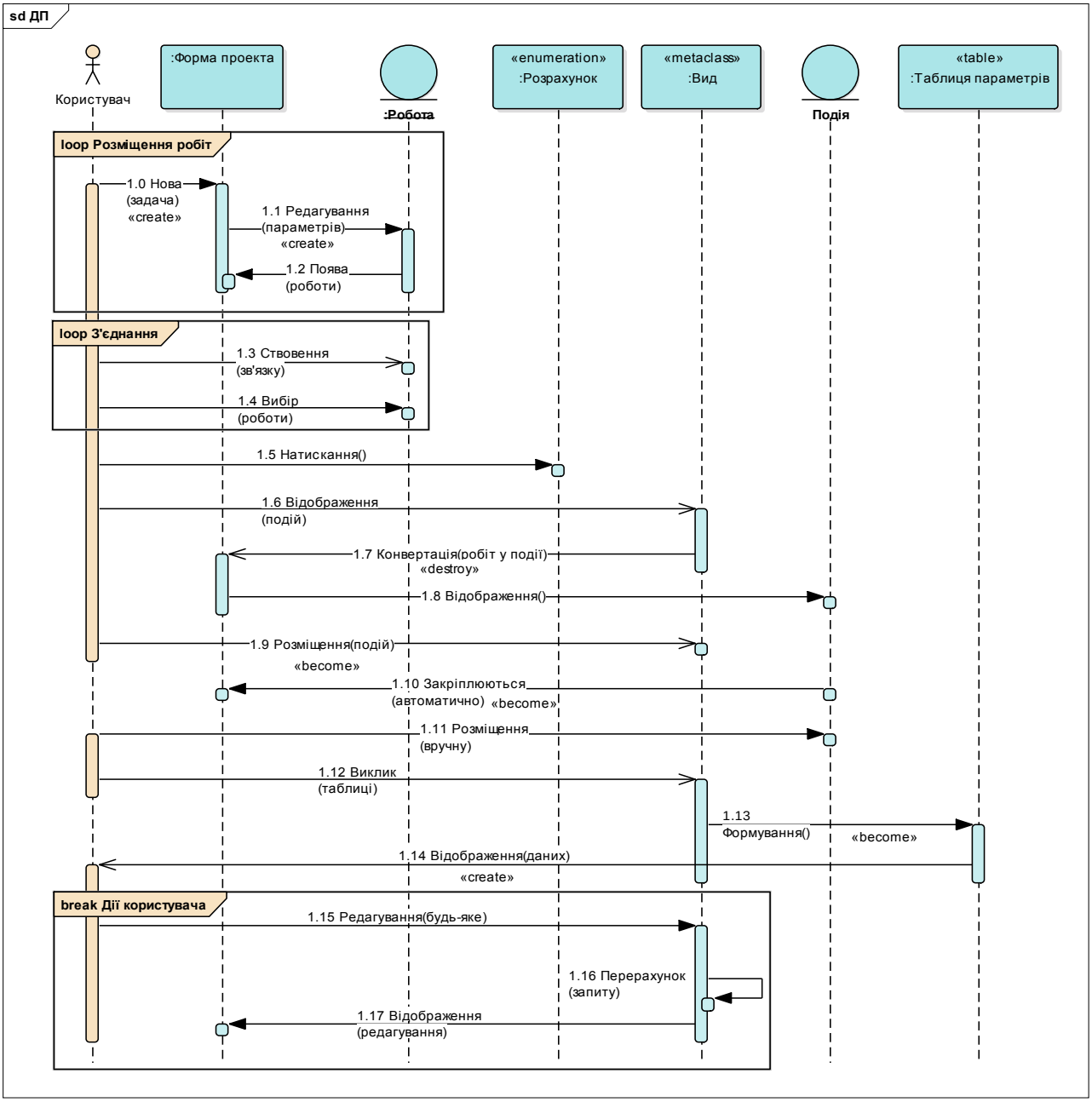

Рис. 4. Діаграма послідовності роботи із ПЗ

Спроектовану діаграму станів ПЗ приведено на рис. 5. Діаграма станів, як видно з рисунку, являє собою каскадну модель задуму створення ПЗ, де у якості поступового каскаду реалізації станів, виступає головна діагональ діаграми з багатьма розгалуженнями (у залежності від точок виходу з конкретних станів) у різні боки.

Діаграми класів використовуються для проектування основного формового наповнення майбутнього ПЗ. Стосовно до нотації UML, класи містять атрибути (власно інкапсульовані дані різного походження) та операції (дії на цими або іншими даними).

Кожні конкретні атрибути та операції класів, що спроектовані у поданому ПЗ подані на рис. 6. Крім класів на діаграмі також важливими для аналізу є зв'язки або відношення (це більш точне визначення стосовно до ДК). Взагалі, сутностей або відношень, щодо методології діаграмі класів – існує багате розмаїття, проте зупинимося лише на тих, що присутні у спроектованій діаграмі класів ПЗ (рис. 6) та потребують додаткових роз'яснень.

Центральним класом на діаграмі – є клас-інтерфейс ("interface"), що являє собою графічний інтерфейс користувача ПЗ. Цей клас першим постає перед користувачем у вигляді завантажувального примітивного класу (primitive) "Форма проекту", що

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2019. No. 2 (8)

має з класом "Інтерфейс" композиційну залежність та визначає первинні геометричні розміри вікна інтерфейсу. Також перед користувачем з'явиться панель меню, елементи якої – є атрибутами класу "Інтерфейс".

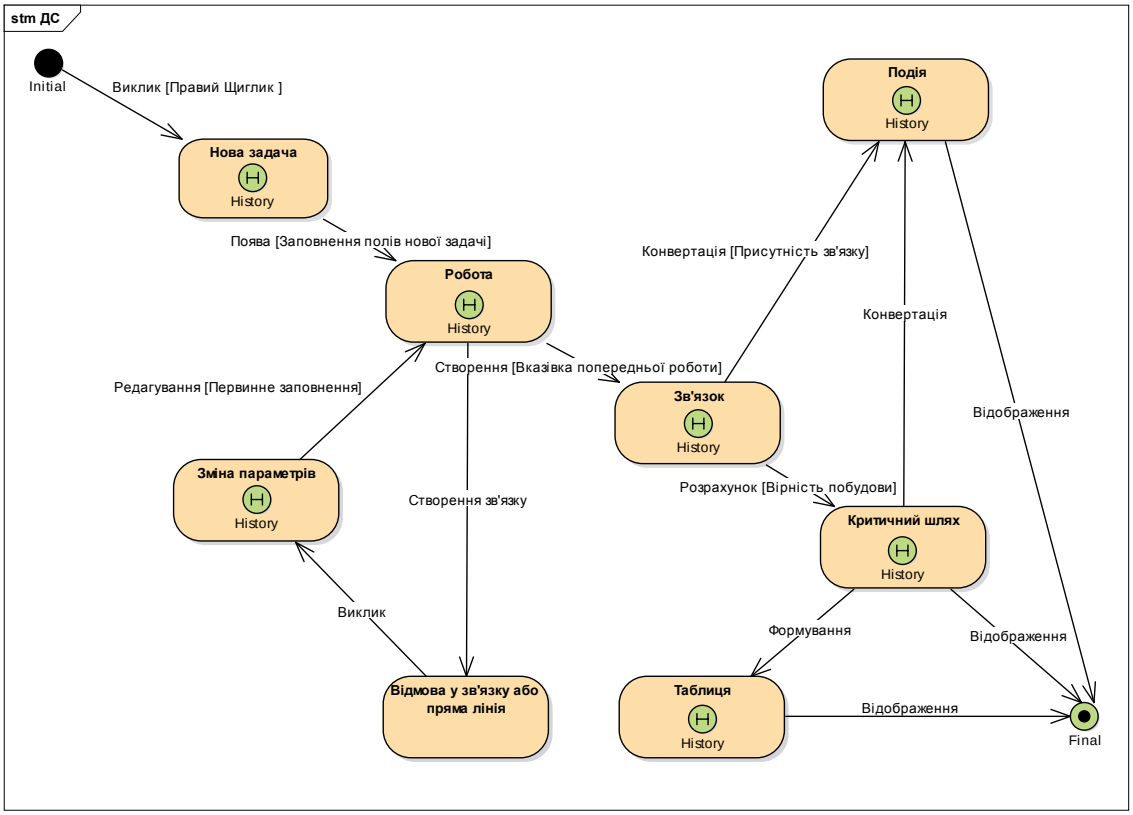

Рис. 5. Діаграма станів для ПЗ

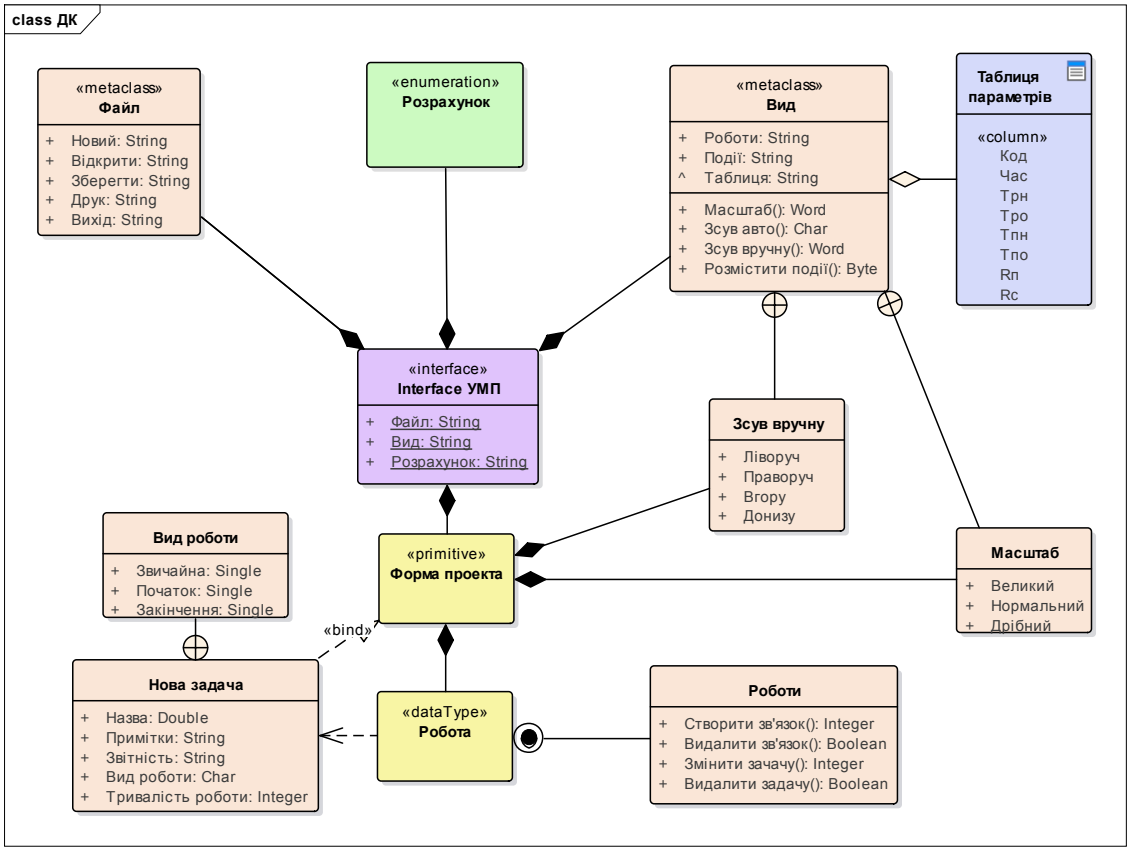

Рис. 6. Діаграма класів ПЗ

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 2 (8) ISSN 2524-2296 (online)

Клас підрахунків (enumeration) "Розрахунок", пов'язаний композиційною залежністю із класом "Інтерфейс", виконує реалізацію чисельних (табличні параметри представлення) та графічних (критичний шлях) параметрів МГ.

Клас "Нова задача", що прив'язано (відношення "Bind") до "Форми проекту", призначено для створення нового та наступного редагування існуючого типу даних (Data Type) "Робота", що являє собою графічне зображення сутності "Робота".

Вкладений (nesting) клас "Вид роботи" – вкладено до класу "Нова задача". Він формує види робіт, що відрізняються наявністю або відсутністю вхідних та вихідних зв'язків.

Діаграма компонентів призначена для вивчення складу компонентів майбутнього ПЗ та вказівки послідовності компіляції та збірки окремих модулів. Головною вимогою до діаграми компонентів – стандартно висунуто – відсутність циклів, тобто послідовність компонентів повинна бути чіткою та прозорою. Програміст працює із компонентами, які можуть бути йому досяжні – у зворотному порядку.

Спроектована діаграма компонентів ПЗ приведена на рис. 7. Розглянемо детальніше її склад.

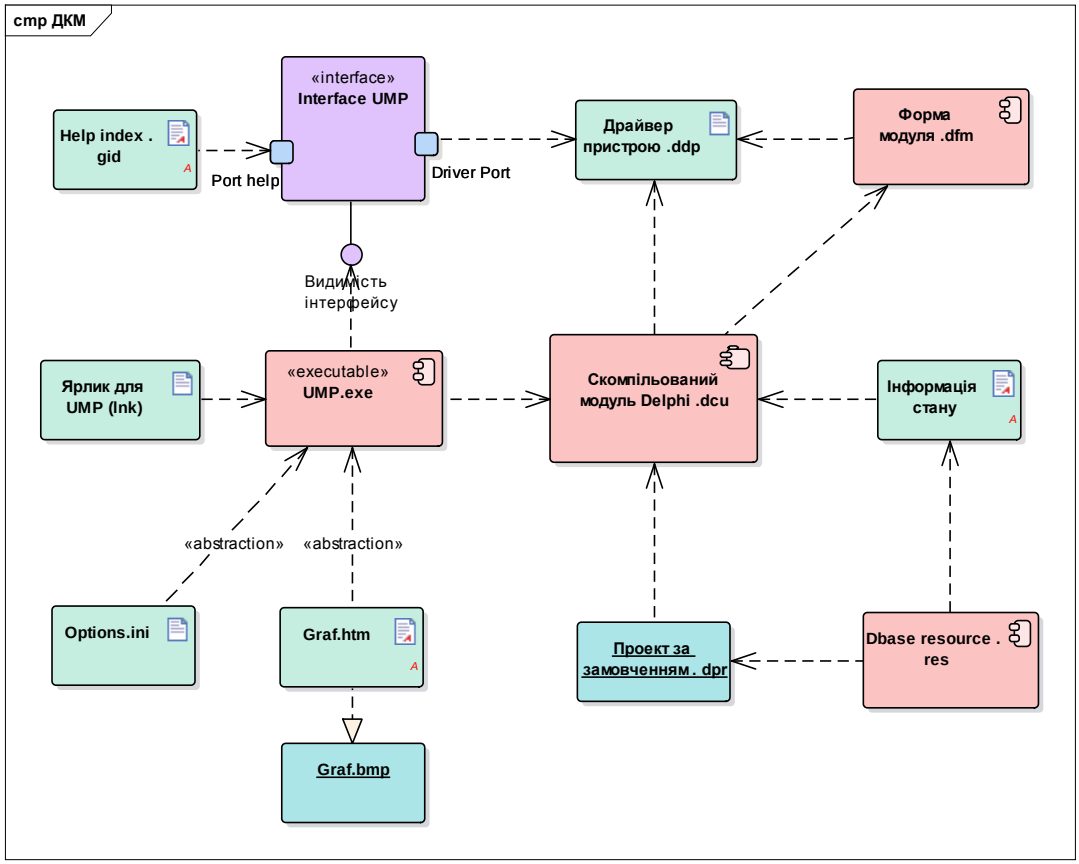

Рис. 7. Діаграма компонентів

Першим компонентом, із яким співпрацює програміст є засіб ідентифікації (artifact) або ярлик виклику "Ярлик для UMP . lnk", що прикріплено до виконавчого файлу (executable) "UMP . exe", що запускає новий проект, через відкритий компонент інтерфейсу (exposed interface) "Видимість інтерфейсу".

Цей компонент служить входом до самого інтерфейсу "Interface UMP", що містить порти допомоги (Porthelp) та драйверу (Port Driver). До порту допомоги (Port help) підключено контекстну допомогу (document artifact) "Helpindex . gid", яка може бути опційно викликана користувачем при виниканні складної ситуації із проектом. Через порт драйвера (Port Driver) – підключено об'єкт (artifact), що містить спеціальні дані – Device Driver Profile "Драйвер пристрою . ddp". Від цього драйверу й залежить вигляд та підтримка розрішувальної здатності усього інтерфейсу ПЗ.

Також від "Драйвер пристрою . ddp" залежать компонент (component) форми Delphi-модуля "Форма модуля . dfm" та пакетний компонент (packaging component) скомпільованого модуля Delphi "Скомпільований модуль . dcu", що також залежить від "Форма модуля . dfm". Таким чином, пакетний компонент "Скомпільований модуль . dcu" має подвійну впорядкованість (рис. 7).

Пакетний компонент "Скомпільований модуль . dcu" виступає головним для цілої низки компонентів, а саме для:

- виконавчого файлу (executable) "UMP . exe", який описано вище;
- службового протоколу (document artifact) "Інформація стану", що містить технічну інформацію щодо поточного та останнього стану проекту;
- об'єкту (object) "Проект за замовченням . dpr", який завантажується у вигляді шаблону при створенні

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2019. No. 2 (8)

нового проекту та який (при необхідності) може бути змінено та доповнено.

Компонент (component) накопичувальної бази даних "Dbase resource . res" має подвійну впорядкованість та компілюється кожного разу при додаванні або зміні інформації у службовому протоколі (document artifact) "Інформація стану" та об'єкту (object) "Проект за замовченням . dpr".

Виконавчий файл (executable) "UMP . exe" є головним для:

- засобу ідентифікації (artifact) або ярлика виклику "Ярлик для spu . lnk", що описано вище;
- службового протоколу (document artifact) опцій та параметрів ПЗ "Options . ini", що містить технічну інформацію щодо останнього геометричного розміру форми проекту, встановленої користувачем;
- браузерної сторінки (Web-document artifact) "Graf . htm", яка відображує остаточну сформовану топологію розміщення задач на МГ; причому браузерна сторінка (Web-document artifact) "Graf . htm" формує (реалізує) само графічне зображення (object) "Graf . bmp" у вигляді самостійного (автономного) bmp-файлу, що може бути як збережено (надруковано, відправлено тощо), так і відредаговано.

Причому, службовий протокол (document artifact) "Options . ini" та браузерна сторінка (Web-document artifact) "Graf . htm" впорядковані абстрактними (abstraction) залежностями (dependency), що означають створення них самих, як сутностей, тільки за потреби користувача та із неодмінною його участю (внесення змін розміру, команди друку чи Web-конвертації.

Екранні знімки спроектованих мережевих графіків планування реінжинірингу програмного забезпечення із розрахунками критичного шляху за допомогою розробленого програмного засобу наведені на рис. 8 (для випадку "вершини-роботи") та на рис. 9 (для випадку "вершини-події").

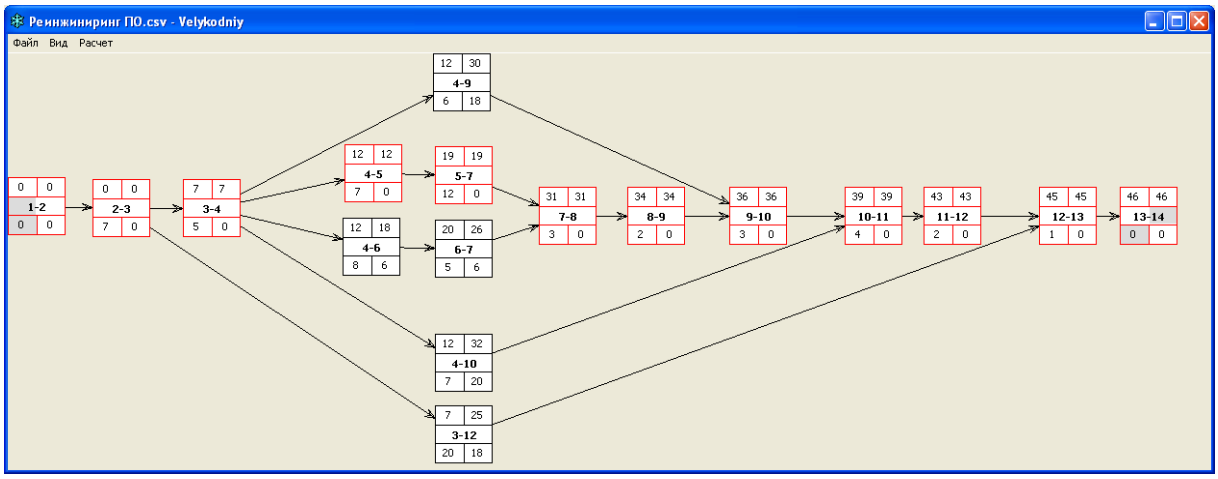

Рис. 8. Мережевий графік у вигляді "вершини-роботи" із розрахованим критичним шляхом

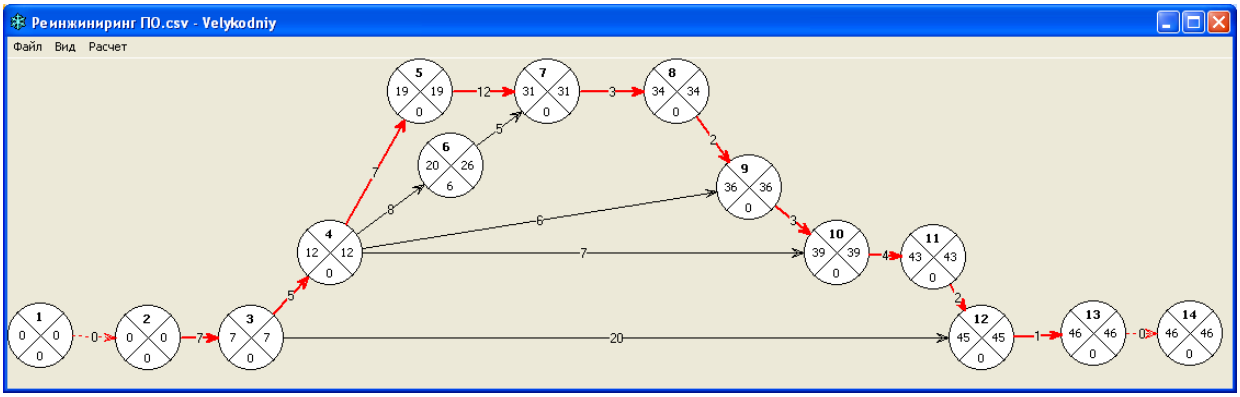

Рис. 9. Мережевий графік у вигляді "вершини-події" із розрахованим критичним шляхом

Обговорення

При аналізі та обговоренні результатів проекту, системними архітекторами було висловлено багато думок щодо засобів реалізації ПЗ. Ці думки стосувалися технологій та мови реалізації, кількості задіяних кодувальників та тестувальників тощо, проте все це складові вже нового етапу реалізації ПЗ. На теперішньому етапі, незалежно від обраних засобів, мов та технологій кодування ПЗ, його архітектура вже є розробленою та готовою. При обранні мови програмування, знадобиться тільки переробка ДКМ.

Висновки та перспективи подальшого розвитку

Слід зазначити, що хоча у сучасних платних спеціалізованих пакетах комп'ютерних програм планування та оперативного управління, в основному, використовується тип графіків "вершини-роботи" – створений ПЗ здатен працювати з усіма типами

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 2 (8) ISSN 2524-2296 (online)

мережевих графіків із можливостями їх всебічної трансформації.

Результатом статті є проекті рішення, що запропоновані авторами. Зміст проектної частини визначається, по-перше, специфікою планування реінжинірингу програмних проектів, по-друге, особливостями конкретних технічних пропозицій до проекту, що піддається управлінню.

У поданій статті спроектована архітектура (проектний "каркас") програмного засобу для управління мережевим плануванням реінжинірингу програмного проекту. Архітектура розроблена у вигляді декількох діаграм різної природи, виконаних із дотриманням нотації UML 2.5 із використанням CASE-інструментарію Enterprise Architect 14. В основу розробки закладені методи мережевого планування за методологію PERT та використання елементів теорії графів. Чисельна та часова оцінка параметрів планування будується за даними, що отримані за методом діаграм Ганта, як облікового для здійснення управління програмними проектами.

Перспективи створення цього ПЗ полягають у кодуванні продукту для кінцевого галузевого користувача (фахівця з управління програмним проектом), якому важливо знати лише послідовність робіт з реінжинірингу програмної системи та тривалість кожної із стадій, і не має особливого значення, яким способом сформований графік, тобто якого він типу, оскільки сам тип мережевого графіку може бути взаємно переконвертований.

Для роботи зі створеним ПЗ буде створено ПМК, у складі якого буде розроблено інструкції користувача із застосування необхідного програмного забезпечення, що доповнені коментарями, відносно роботи спроектованого ПЗ.

Список літератури

1. Program Evaluation and Review Technique (PERT). URL : https://www.inc.com/encyclopedia/program-evaluation-and-review-technique-pert.html (дата звернення : 01.05.2019).
2. Bondy J. A., Murty U. S. R. Graph Theory. San Francisco : Springer, 2008. 655 p. DOI: https://doi.org/10.1007/978-1-84628-970-5.
3. Емеличев В. А., Мельников О. И., Сарванов В. И. Лекции по теории графов. Изд. 2. Москва : Наука, 2009. 392 с.
4. Jungnickel D. Graphs, Networks and Algorithms. Fourth Edition. Berlin : Springer, 2013. 677 p. DOI: https://doi.org/10.1007/978-3-642-32278-5.
5. Sedgewick R. Algorithms in Java, Third Edition, Part 5: Graph Algorithms. Boston : Addison Wesley, 2003. 528 p.
6. Коммерческие специализированные программы для построения графов. URL : http://www.boost.org/ (дата звернення : 02.04.2019).
7. LION Graph Visualizer. URL :http://lion.disi.unitn.it/intelligent-optimization/visualizer.html (дата звернення : 03.04.2019).
8. Графоанализатор – среда визуализации графов. URL :http://grafoanalizator.unick-soft.ru/ (дата звернення : 03.04.2019).
9. Jalote P. Software project management in practice. Indianapolis: Addison-Wisley, 2005. 242 p.
10. Великодний С. С. Метод представлення оцінки реінжинірингу програмних систем за допомогою проектних коефіцієнтів. *Сучасний стан наукових досліджень та технологій в промисловості*. 2019. № 1 (7). С. 34–42. DOI: https://doi.org/10.30837/2522-9818.2019.7.034.
11. Punmia B. C., Khandelwal K. K. Project Planning and Control with PERT and CPM.New Delhi :Laxmi Publications, 2016. 258 p.
12. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Eight Edition. New Jersey : John Wiley & Sons, 2003. 914 p.
13. Великодний С. С., Тимофєєва О. С., Зайцева-Великодна С. С. Метод розрахунку показників оцінки проекту при виконанні реінжинірингу програмних систем. *Радіоелектроніка, інформатика, управління*. 2018. № 4. С. 135–142. DOI: https://doi.org/10.15588/1607-3274-2018-4-13.
14. Великодний С. С. Реінжиніринг систем моніторингу та дистанційного управління судновими енергетичними установками. XXII Міжн. конф. з автом. управл. "Автоматика 2015", 10–11 вер., 2015. Одеса. С. 133–134.
15. Невлюдов И. Ш., Великодный С. С., Омаров М. А. Использование CAD/CAM/CAE/CAPP при формировании управляющих программ для станков с ЧПУ. Восточно-Европейский журнал передовых технологий. 2010. Т. 2. Вып. 2 (44). С. 37–44.

References

1. "Program Evaluation and Review Technique (PERT)", available at : https://www.inc.com/encyclopedia/program-evaluation-and-review-technique-pert.html (last accessed 01.05.2019).
2. Bondy, J. A., Murty, U. S. R. (2008), *Graph Theory*. Springer, San Francisco, 655 p. DOI: https://doi.org/10.1007/978-1-84628-970-5.
3. Emelichev, V. A., Mel'nikov, O. I., Sarvanov, V. I. (2009), *Lectures on graph theory. 2nd ed.* [*Lektsii po teorii grafov. Izd. 2*], Moscow, 392 p.
4. Jungnickel, D. (2013), Graphs, Networks and Algorithms. 4th ed., Springer, Berlin, 677 p. DOI: https://doi.org/10.1007/978-3-642-32278-5.
5. Sedgewick, R. (2003), *Algorithms in Java, 3rd ed., Part 5: Graph Algorithms*, Addison Wesley, Boston, 528 p.
6. "Commercial specialized programs for graph construction" ["Kommercheskie spetsializirovannye programmy dlya postroeniya grafov"], available at : http://www.boost.org/ (last accessed 02.04.2019).
7. "LION Graph Visualizer", available at :http://lion.disi.unitn.it/intelligent-optimization/visualizer.html (last accessed 03.04.2019).
8. "Graph analyzer – graph visualization environment" ["Grafoanalizator – sreda vizualizatsii grafov"], available at :http://grafoanalizator.unick-soft.ru/ (last accessed 03.04.2019).
9. Jalote, P. (2005), *Software project management in practice*, Addison-Wisley, Indianapolis, 242 p.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2019. No. 2 (8)

10. Velykodniy, S. (2019), "Method of presenting the assessment for reengineering of software systems with the project coefficients help" ["Metod predstavlennia otsinky reinzhynirynhu prohramnykh system za dopomohoiu proektnykh koefitsiientiv"], Innovative Technologies And Scientific Solutions For Industries, No. 1 (7), P. 34–42. DOI: https://doi.org/10.30837/2522-9818.2019.7.034
11. Punmia, B. C., Khandelwal, K. K. (2016), *Project Planning and Control with PERT and CPM*. Laxmi Publications, New Delhi, 258 p.
12. Kerzner, H. (2003), Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 8ht ed., John Wiley & Sons New Jersey, 914 p.
13. Velykodniy, S. S., Tymofieieva, O. S., Zaitseva-Velykodna, S. S. (2018), "The calculation method for indicators project estimation in the implementation of software systems re-engineering" ["Metod rozrakhunku pokaznykiv otsinky proektu pry vykonanni reinzhynirynhu prohramnykh system"], *Radio Electronics, Computer Science, Control*, No. 4, P. 135–142. DOI: https://doi.org/10.15588/1607-3274-2018-4-13.
14. Velykodniy, S. (2015), "Reengineering of SCADA-systems by shipping energy plants ["Reinzhiniring sistem monitoringu ta distantsiynogo upravlinnya sudnovimi energetichnimi ustanovkami"], *22th International Conference "Automatic 2015", 10–11 sep. : proceedings*, Odessa, P. 133–134.
15. Nevlyudov, I. Sh., Velykodniy, S. S., Omarov, M. A. (2010), "Using CAD / CAM / CAE / CAPP when forming control programs for CNC machines" ["Ispol'zovanie CAD/CAM/CAE/CAPP pri formirovanii upravlyayushchikh programm dlya stankov s ChPU"], *Eastern-European Journal of Enterprise Technologies*, Vol. 2, Issue 2 (44), P. 37–44.

Надійшла (Received) 15.05.2019

Відомості про авторів / Сведения об авторах / About the Authors

Великодний Станіслав Сергійович – кандидат технічних наук, доцент, Одеський державний екологічний університет, доцент кафедри інформаційних технологій, Одеса, Україна; e-mail: velykodniy@gmail.com; ORCID: https://orcid.org/0000-0001-8590-7610.

Великодный Станислав Сергеевич – кандидат технических наук, доцент, Одесский государственный экологический университет, доцент кафедры информационных технологий, Одесса, Украина.

Velykodniy Stanislav – PhD (Computer Science), Associate Professor, Odessa State Environmental University, Associate Professor of the Department of Information Technologies, Odessa, Ukraine.

Бурлаченко Жанна Вікторівна – Одеський державний екологічний університет, аспірант кафедри інформаційних технологій, Одеса, Україна; e-mail: 7035373@ukr.net; ORCID: https://orcid.org/0000-0001-8451-5527.

Бурлаченко Жанна Викторовна – Одесский государственный экологический университет, аспирант кафедры информационных технологий, Одесса, Украина.

Burlachenko Zhanna – Odessa State Environmental University, Post-graduate Student of the Department of Information Technologies, Odessa, Ukraine.

Зайцева-Великодна Світлана Сергіївна – Одеський державний екологічний університет, аспірант кафедри інформатики, Одеса, Україна; e-mail: svetlana.zaytseva@gmail.com; ORCID: https://orcid.org/0000-0001-7453-8821.

Зайцева-Великодная Светлана Сергеевна – Одесский государственный экологический университет, аспирант кафедры информатики, Одесса, Украина.

Zaitseva-Velykodna Svitlana – Odessa State Environmental University, Post-graduate Student of the Department of Information Technologies, Odessa, Ukraine.

РАЗРАБОТКА АРХИТЕКТУРЫ ПРОГРАММНОГО СРЕДСТВА ДЛЯ УПРАВЛЕНИЯ СЕТЕВЫМ ПЛАНИРОВАНИЕМ РЕИНЖИНИРИНГА ПРОГРАММНОГО ПРОЕКТА

Предмет исследования – программное средство построения графической сетевой модели реинжиниринга программного проекта. **Цель** исследования – разработка системной архитектуры программного средства для автоматизированного проектирования сетевых графиков организации производства по реинжинирингу программных систем в рамках управления проектами. Наряду с линейными графиками и табличными расчетами сетевые методы планирования находят широкое применение при разработке перспективных планов и моделей создания сложных производственных систем и других объектов долгосрочного использования. **Задачей** перед созданием программного средства является способность работать со всеми типами сетевых графиков с возможностями их всесторонней трансформации. **Методы**. В основу статьи заложена методы сетевого планирования по методологии PERT (Program (Project) Evaluation and Review Technique), использование элементов теории графов и метода диаграмм Ганта, как учетного для осуществления управления проектами. Моделирование системной архитектуры программного обеспечения выполняется в рамках методологии UML (Unified Modeling Language) 2.5 с использованием CASE-инструментария Enterprise Architect 14. **Результаты.** В статье разработано программное средство для управления сетевым планированием реинжиниринга программного проекта. **Выводы.** Архитектура разработана в виде нескольких структурных и поведенческих диаграмм, а именно: диаграмма вариантов использования, иллюстрирующая аналитику детальное представления об области применения программного средства; диаграмма последовательности, предназначенная для формирования представления программиста о порядке выполнения действий при работе с будущим программным средством; диаграмма состояний, необходимая для наглядного представления тех состояний, в которых программное средство может находиться в разные моменты времени; диаграмма классов, использующаяся для проектирования основного формового наполнения будущего программного средства; диаграмма

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 2 (8) *ISSN 2524-2296 (online)*

компонентов, предназначенная для изучения состава компонентов будущего программного средства и указания последовательности компиляции и сборки отдельных модулей. Численная и временная оценка параметров планирования строится по данным, полученные из проектных диаграмм Ганта.

Ключевые слова: управление проектом; граф; сетевой график; программное средство; реинжиниринг; CASE-средство; UML-диаграмма.

ARCHITECTURE DEVELOPMENT OF SOFTWARE FOR MANAGING NETWORK PLANNING OF SOFTWARE PROJECT REENGINEERING

Subject of the research is a software tool for construction of graphic network model of reengineering the software project. **Purpose** of the research is the development of technical architecture of software tool for automated design of network schedules for organization of production the software systems reengineering within the framework of project management. Along with the linear charts and table calculations, network planning methods are extensively used in the development of long-term plans and models for the creation of complex production systems and other objects of the long-term use. **The task** before creating a software tool is the ability to work with all types of network charts with the possibilities of their comprehensive transformation. **Methods.** The article is based on the methods of network planning for the PERT (Program Evaluation and Review Technique) methodology, the use of elements of graph theory and the Gantt chart method as an accounting method for project management. Simulation of the system software architecture is carried out within the UML (Unified Modeling Language) 2.5 methodology using the CASE toolkit Enterprise Architect 14. **Results.** The software architecture for managing network planning of software project reengineering is designed in the article. **Conclusions.** The architecture is developed in the form of several structural and behavioral diagrams, namely: use case diagram, which provides an analyst with a detailed idea of the software field of application; sequence diagram that is designed to create a programmer's imagination on how to perform actions when working with a future program tool; state chart diagram that is required for a visual representation of those states in which the software can be at different times; class diagrams that are used to design the main form filling of the future software; component diagram that is designed to examine the composition of the components of the future software and indicate the sequence of compilation and assembly of individual modules. The numerical and temporal estimation of the planning parameters is based on the data obtained from the Gantt design charts.

Keywords: project management; graph; network schedule; software; reengineering; CASE-tool; UML-diagram.

Бібліографічні описи / Bibliographic descriptions

Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Розробка архітектури програмного засобу для управління мережевим плануванням реінжинірингу програмного проекту. *Сучасний стан наукових досліджень та технологій в промисловості*. 2019. № 2 (8). С. 25–35. DOI: https://doi.org/10.30837/2522-9818.2019.8.025.

Velykodniy, S., Burlachenko, Zh., Zaitseva-Velykodna, S. (2019), "Architecture development of software for managing network planning of software project reengineering", *Innovative Technologies and Scientific Solutions for Industries*, No. 2 (8), P. 25–35. DOI: https://doi.org/10.30837/2522-9818.2019.8.025.