

Лабораторна робота №1

Функціональне моделювання процесів інформаційних систем за допомогою CASE – засобів.

Побудова функціональних моделей із використанням технології IDEF0.
Вартісний аналіз для оцінювання моделей

Серед найзручніших методів аналізу процесів у різноманітних системах, в тому числі, системах обробки інформації є їх моделювання за допомогою технології IDEF0. Під моделлю в IDEF0 вважають опис системи, як графічний, так і текстовий, який дає відповідь на деякі попередньо визначені питання. За допомогою IDEF0 система може бути представлена як сукупність взаємодіючих робіт або функцій. Саме така функціональна орієнтація є принциповою, оскільки функції системи аналізуються незалежно від об'єктів, якими вони оперують.

Моделювану систему розглядають як довільну підмножину необмеженої множини. Система має межу і її взаємодія із оточуючим світом може бути описана як вхід (щось, що обробляється системою), вихід (результат діяльності системи), управління (стратегії і процедури, під управлінням яких відбувається робота) і механізм (ресурси, які необхідні для проведення роботи).

Процес моделювання системи в IDEF0 починають із визначення контексту, тобто, найбільш абстрактного рівня опису системи в цілому. В контекст входить визначення суб'єкту моделювання, мети (purpose) і точки зору (viewpoint) на модель. Під час цих кроків визначають глибину і ширину моделювання. Ширину визначає межі моделі, а саме, що розглядають всередині системи, а, що ззовні. Глибиною визначається рівень деталізації системи. Затрати на побудову моделі ростуть в геометричній прогресії відповідно до глибини деталізації аналізованої системи. Після визначення меж додаткові об'єкти не повинні вноситися в модель, оскільки всі об'єкти системи взаємозв'язані. Мета повинна давати відповідь на питання: чому процес має бути змодельований, що повинна представити модель, що може отримати споживач моделі? Точка зору на моделювану систему має відповідати меті моделювання і відображати єдину точку зору на модель. Для представлення альтернативних точок зору використовують графічні побудови Тільки для експозиції (FEO – for exposition only), які представляють собою “відбитки” векторних діаграм. Опис моделі, в цілому, задають через меню Model/Model Properties.

Зокрема, на вкладці Status можна описати статус моделі, наприклад, чорновик, робочий, готовий до опублікування. Вкладка Source дозволяє описати джерела, які використовують під час моделювання. На вкладці General можна задати часові рамки варіанту моделі: сучасний AS IS і в майбутньому TO BE.

Основу методології IDEF0 складає графічна нотація опису функціональних процесів, а модель складається із ієрархічної сукупності взаємозв'язаних діаграм, які розміщені на окремих листах. Модель може містити 4 типи діаграм: контекстна (context) (єдина у кожній моделі), діаграма декомпозиції (decomposition (child)), діаграма дерева вузлів (node tree), діаграма тільки для експозиції (feo).

Контекстна діаграма є вершиною деревовидної структури діаграм і подає найбільш загальний опис функціональної системи і її зовнішні взаємодії. Діаграми декомпозиції представляють собою крупні фрагменти, які утворюються у результаті функціонального аналізу і розбиття системи. Така процедура триває до досягнення потрібного рівня деталізації процесів. Діаграма дерева вузлів показує ієрархічну

залежність процесів, а не їхні взаємозв'язки. Таких діаграм в моделі може бути скільки завгодно і вони можуть бути побудовані на різну глибину.

Нотація (синтаксис) IDEF0 є однаковою для всіх діаграм в моделі і включає в себе наступні елементи:

- **Роботи** (Activity) – представлені на діаграмі прямокутником і позначають іменовані процеси, завдання або функції, які відбуваються протягом певного часу і мають розпізнаванні результати. Всі роботи мусять бути названі, наприклад, “виготовлення деталі”, “форматування документу” і визначені їх властивості через контекстне меню.
- **Стрілки** (Arrow) – описують взаємодію робіт між собою, а також із зовнішнім світом (останні, це, так звані, граничні стрілки (border arrows)). Стрілки представляють будь-яку інформацію, яка рухається в системі між процесами обробки або отримується ззовні і є результатом обробки. Стрілки повинні бути названими, наприклад, “виріб”, “оцінка”, “деталь” тощо за допомогою контекстного меню властивостей Arrow properties. Імена стрілок автоматично заносяться в словник Arrow Dictionary і в ньому можна їх додатково описати, зокрема, дати формальне визначення.

В IDEF0 розрізняють п'ять типів стрілок:

- **Вхід** (Input) – інформація або матеріал, які використовуються або перетворюються роботою для досягнення результату (виходу (output)). Стрілку входу рисують до лівої грані прямокутника. У стандарті допускається, що робота може не мати стрілки на вході, бажано не більше трьох.
- **Управління** (Control) – правила, стратегії, процедури або стандарти, які керують роботою. Кожна робота повинна мати хоча б одну стрілку управління, яку рисують до верхньої грані прямокутника. Управління впливає на роботу, але нею не перетворюється.
- **Вихід** (Output) – матеріал або інформація, які отримують в результаті виконання роботи. Стрілка виходить із правої грані прямокутника. Робота повинна мати хоча б одну стрілку виходу, тому що, робота без результату є беззмисловою і не моделюється.
- **Механізм** (Mechanism) – ресурси, які виконують роботу, наприклад, персонал, прикладна програма, пристрій і т.п. Стрілку рисують, як таку, що входить у нижню грань прямокутника. Згідно стандарту стрілки механізму можуть не зображуватися в моделі.
- **Виклик** (Call) – стрілка, яка може виходити із нижньої грані роботи, і, яка вказує на існування іншої моделі роботи. Таку стрілку використовують для позначення того, що деяка робота виконується поза межами модельованої системи. В програмі VPwin стрілки виклику застосовуються для розщеплення і злиття моделей.

Напрямок стрілок визначається порядком їх рисування, крім того, слід розрізняти граничні і внутрішні стрілки. Граничні стрілки рисують, активізуючи появу чорної смуги, після наведення курсора на полях діаграми до (від) відповідних граней робіт. Візуальну прив'язку назв до стрілок, формат тексту, обрізку довжини, стиль можна реалізувати за допомогою контекстного меню (команди Squiggle, Font, Trim, Style).

ICOM коди – автоматично ідентифікують стрілки діаграми (аббревіатура від Input, Control, Output, Mechanism). Код містить префікс I, C, O або M і

порядковий номер. Для відображення ICOM кодів необхідно виставити опцію у властивостях моделі на вкладці Display.

Побудова зв'язків робіт.

Після декомпозиції діаграми контекстного рівня всі пов'язані із нею стрілки, окрім, стрілки виклику, появляються на дочірній діаграмі у незв'язаному вигляді (unconnected border arrows). Якщо їх не зв'язати із роботами декомпонованої діаграми, то це призведе до появи синтаксичної помилки моделі. У цьому випадку зв'язки слід організовувати вже не від границі діграми, а від дочірніх стрілок, крім того, на діаграмі декомпозиції використовують, так звані, внутрішні стрілки для зв'язку робіт між собою.

В IDEF0 розрізняють п'ять типів зв'язків робіт:

- Зв'язок по входу (output-input) – у цьому випадку вихід попередньої роботи пов'язаний із входом наступної;
- Зв'язок по управлінню (output-control) – у цьому випадку вихід попередньої роботи керує наступною, тобто, дані або матеріали не підлягають обробці і вихідна робота є домінуючою;
- Зворотній зв'язок по входу (output-input feedback) – у цьому випадку вихід наступної роботи скеровують на вхід попередньої, такий тип використовують для відтворення циклів;
- Зворотній зв'язок по управлінню (output-control feedback) – у цьому випадку вихід наступної роботи скеровують на управління попередньою, такий тип використовують для аналізу ефективності процесів обробки;
- Зв'язок вихід-механізм (output-mechanism) – у цьому випадку вихід роботи скеровують на механізм іншої, такий тип зв'язку показує, що одна робота підготовлює ресурси необхідні для виконання іншої.

Згідно із стандартом стрілки можуть бути як явними, тобто, такими, що мають у якості джерела одну роботу, так і приймача одну роботу, так і такими, що розгалужуються і зливаються. Стрілки, що розгалужуються представляють інформацію або об'єкти, породжені однією роботою і, які можуть використовуватися зразу у декількох інших роботах. З іншого боку, результати породжені в різних роботах, можуть представляти собою однакові або однорідні дані, інформацію, які в подальшому використовують або переробляють в єдиному місці. У цьому випадку застосовують стрілки, що зливаються. При цьому, зміст кожної із стрілок, що галузяться або зливаються, позначається іменуванням її сегментів. Правила найменування таких стрілок полягають у тому, що, наприклад, якщо стрілка іменована до розгалуження, а після розгалуження ні один із сегментів не найменований, то вважається, що кожна гілка моделює ті ж дані і об'єкти, що і до розгалуження. Якщо ж лише якась із гілок не є іменованою, то вона моделює ті ж дані і об'єкти, як і до розгалуження. Недопустимою є ситуація (синтаксична помилка), якщо стрілка до розгалуження немає назви і після розгалуження немає назви одна із гілок. Правила іменування стрілок, що зливаються є аналогічними.

Тунелювання стрілок.

Заново внесені граничні стрілки на діаграмі декомпозиції, а не в контекстній діаграмі, зображаються програмою в квадратних дужках і автоматично не появляються на діаграмі верхнього рівня. Команда контекстного меню Resolve

Border Arrow реалізовує міграцію такої стрілки на діаграму верхнього рівня, а команда Change to Resolved Rounded Tunnel – тунелює стрілку і вона не попадає на іншу діаграму. Тунельна стрілка зображається з круглими дужками на кінці.

Тунелювання застосовують для зображення незначних стрілок. Для того, щоб малозначима стрілка, тобто стрілка, яка представляє малозначимі об'єкти або дані, не зображалась на всіх рівнях моделі і не ускладнювала прочитання діаграм, можна тунелювати стрілку на якомусь із рівнів. Таке тунелювання називається “не в батьківській діаграмі”. Іншим випадком тунелювання є випадок міграції стрілки механізму з верхнього рівня на нижній, якщо на цьому рівні такий механізм використовують для всіх робіт без винятку. Тому на нижньому рівні стрілка механізму може бути видалена, а на батьківській діаграмі (parent diagram) її можна тунелювати, а коментарях до стрілки або в словнику вказати, що механізм буде використано у всіх роботах дочірньої діаграми декомпозиції. Таке тунелювання називають “не в дочірній роботі”.

Декомпонувати роботу на одну роботу немає змісту, діаграми декомпозиції з кількістю робіт більше 8 погано читаються і це межа закладена у стандарт, раціонально застосовувати від 3 до 6 блоків робіт у декомпонованій діаграмі. Порядок домінування робіт на діаграмі у часі і важливості їх виконання з верхнього лівого до правого нижнього кута. Окрім того, що порядок домінування робіт показує їх взаємозв'язок, він мінімізує вигини і перетини стрілок. Програмно можна забезпечити оптимальні відстані між стрілками за допомогою Властивостей моделі (Model Properties) вкладка Layout опція Automatically Space Arrows. Керувати нумерацією робіт можна на вкладці Numbering. Нумерація відображається автоматично в правому нижньому куті блоку роботи, а недекомпонована робота позначається діагональною рисою у лівому верхньому куті блоку. Крім того, на блоці можуть бути позначення, які використовують для вартісного аналізу робіт, так званого, ABC (Activity Based Costing). Зворотні зв'язки по входу відображають нижньою петлею, а по управлінню – верхньою. Циклічні зворотні зв'язки можна використовувати тільки у особливих випадках, тоді коли підкреслюють значення повторно використововуваного об'єкту. Як правило, такі зв'язки зображають через діаграму декомпозиції. Щоб представити циклічний зворотній зв'язок слід спочатку побудувати звичайний зв'язок по виходу і застосувати його розгалуження до входу роботи, а непотрібну гілку вилучити. В цілому, в процесі моделювання слід мінімізувати кількість перетинів, петель і поворотів на діаграмах.

Побудова каркасу діаграми, дерева вузлів і FEO діаграм.

Каркас містить заголовок і підвал. Заголовок використовують для відслідковування діаграми в процесі моделювання. Нижня частина призначена для ідентифікації і позиціонування в ієрархії діаграми.

Поля заголовку мають наступний зміст:

Used at – використовують для вказівки на батьківську роботу у випадку, якщо на поточну діаграму посилались за допомогою стрілки виклику;

Author, Date, Rev, Project – авторство, дата, ім'я проекту, дата останнього редагування діаграми;

Notes 12345678910 – використовують під час проведення сеансу експертизи, експерт закреслює відповідну цифру під час внесення зауваження до проекту;

Working – нова діаграма, кардинально оновлена діаграма або новий автор діаграми;

Draft – діаграма пройшла первинну експертизу;

Recommended – діаграма пройшла експертизу і ніяких доповнень не очікується;

Publication – діаграма готова до кінцевої публікації;

Reader – ім'я читача (експерта);

Context – схема розміщення робіт на діаграмі верхнього рівня.

Поля підвалу мають наступний зміст:

Node – номер вузла діаграми (номер батьківської роботи);

Title – ім'я діаграми (за замовчування ім'я батьківської роботи);

Number – C-Number (унікальний номер версії діаграми);

Page – номер сторінки.

Значення полів каркасу задають за допомогою властивостей діаграми Diagram Properties.

Використання діаграм дерева вузлів.

Оскільки процедура створення моделі функціональних процесів є, як правило, ітераційною, наприклад, роботи змінюють ієрархію в системі декомпозиції діаграм, то для того, щоб не загубитися у моделі і перевірити спосіб декомпозиції, рекомендують після кожних змін у діаграмах моделі створювати діаграму дерева вузлів. Створити дерево вузлів можна через меню Diagram/Add Node Tree і варто задати ім'я дерева, оскільки при побудові декількох дерев і використанні одного і того ж кореня їх можна буде розрізнити по імені. В діалоговому вікні можна задати вигляд дерева, зокрема, наявність діагональних або ортогональних ліній, відображення найнижчого рівня моделі у вигляді списку або прямокутників (Bullet Last Level) і т.д.

Використання FEO діаграм.

Для відображення окремих поглядів на модель або деталей моделі, які не підтримуються синтаксисом IDEF0, можна використати FEO діаграми, оскільки вони не підпадають під синтаксичний аналіз. Створити FEO діаграми можна через меню Diagram/Add FEO diagram, вказуючи ім'я діаграми FEO і тип та ім'я батьківської діаграми. Нова діаграма отримує номер, який генерується автоматично.

Забезпечення колективної роботи над проектом.

Можливість колективної роботи над проектом можна реалізувати забезпечуючи злиття і розщеплення моделей. Окрема гілка може бути використана для використання в якості незалежної моделі, для доопрацювання або архівування. Програма VPwin використовує для злиття і розщеплення стрілки виклику.

Для реалізації злиття необхідно виконати наступні вимоги:

- Обидві моделі, які об'єднують повинні бути відкриті в програмі;
- Ім'я моделі - джерела, яку приєднують до моделі – мети повинно співпадати з іменем стрілки виклику в моделі – меті;
- Стрілка виклику має виходити із недекомпонованої роботи;

- Ім'я контекстної роботи в моделі – джерелі і роботи в моделі – меті, до якої під'єднують модель – джерело, мають співпадати;
- Модель – джерела має містити хоча б одну діаграму декомпозиції.

Для злиття моделей вибирають у контекстному меню стрілки виклику в моделі – меті команду Merge Model.

Під час об'єднання моделей також відбувається об'єднання словників стрілок і робіт. Якщо існують однакові визначення, то можливим є перезапис визначень або прийняття визначень із моделі – джерела.

Після підтвердження злиття моделей модель – джерело приєднується до моделі – мети, стрілка виклику зникає, а робота з якої виходила стрілка виклику стає декомпонованою, тобто, до неї приєднується діаграма декомпозиції першого рівня моделі – джерела. Стрілки, які стосуються роботи на діаграмі моделі – мети, автоматично не мігрують в декомпозицію, а відображаються як нерозв'язані (unresolved). Їх треба тунелювати за вище наведеним алгоритмом.

У процесі злиття моделей модель – джерело залишається незмінною, а до моделі – мети насправді під'єднується її копія. Якщо в подальшому модель – джерело буде редагуватися, то ці зміни автоматично не попадуть у відповідну гілку моделі – мети. Розщеплення моделей відбувається аналогічно. Для відщеплення гілки від моделі необхідно в контекстному меню роботи, яка є вже декомпонованою вибрати команду Split Model. В діалоговому вікні Split Options необхідно вказати ім'я моделі, яку створюють. Після підтвердження розщеплення в старій моделі робота стає недекомпонованою, до неї програмою створюється стрілка виклику, причому її ім'я буде співпадати з іменем нової моделі, і також буде створено нову модель, у якій ім'я контекстної роботи буде співпадати з іменем роботи, від якої була відірвана декомпозиція.

Звіти в програмі BPwin.

Звіти в моделі можна згенерувати командою Tools – Model Report. Всього є 7 типів звітів:

1. Model Report – звіт включає в себе інформацію про контекст моделі, а саме: ім'я моделі, точку зору, мета, ім'я автора, дату створення і т.д.
2. Diagram Report – звіт щодо конкретної діаграми, включає в себе список об'єктів (робіт, стрілок, сховищ даних, зовнішніх посилань і т.д.)
3. Diagram Object Report – найбільш повний звіт щодо моделі. Може містити повний список об'єктів моделі (робіт, стрілок з вказуванням їх типу і т.д.), а також властивості визначені користувачем.
4. Activity Cost Report – звіт про результати вартісного аналізу.
5. Arrow Report – звіт щодо стрілок. Може містити інформація із словника стрілок, інформацію про роботу – джерело, роботу – до якої напрямлена стрілка, і інформацію щодо розгалуження і злиття стрілок.
6. Data Usage Report – звіт щодо результатів зв'язування моделі - процесів і моделі – даних.
7. Model Consistency Report – звіт щодо синтаксичних помилок моделі.

Синтаксичні помилки IDEF0 поділяють на три типи:

- По-перше, помилки, які програма виявити не в стані. Наприклад, синтаксис IDEF0 вимагає, щоб ім'я роботи представлялось у формі дії, а імена стрілок були іменниками. Програма не аналізує синтаксис природної мови в зміст імен об'єктів, а тому, звичайно, ігнорує їх.

- Помилки другого типу є недопустимими в програмі, наприклад, кожна із граней допускає побудову стрілок тільки певного типу.
- Третій тип помилок програма дозволяє пропустити, але, одночасно, їх детектує. Такий список помилок може містити, неіменовані роботи і стрілки (unnamed arrow, unnamed activity), незв'язані стрілки (unconnected border arrow), нерозв'язані стрілки (unresolved (square tunneled) arrow connections, роботи, які не мають хоча б однієї стрілки управління або виходу (activity has no control, activity has no output).

Під час вибору пункту меню, який відповідає якому-небудь звіту, появляється діалог налаштування звіту.

Як правило, в процесі побудови функціональних моделей першою будують модель існуючих процесів, так звана, модель AS-IS. Після побудови первісної моделі функціональних процесів потоки даних і об'єктів можуть бути перенаправлені і покращені, якщо у цьому виникає потреба, і, як результат, можна побудувати модель або декілька моделей TO-BE. Для того, щоб проаналізувати якість створюваних моделей, з огляду на ефективність описуваних процесів, необхідною є кількісна оцінка моделей.

ВРwin надає аналітику процесів в інформаційних системах два інструменти для оцінки моделі – вартісний аналіз, який базується на роботах (Activity Based Costing, ABC), і властивості, визначені користувачем (User Defined Properties, UDP).

Функціонально вартісний аналіз.

ABC є розповсюдженою методикою, яка використовується у практиці аналізу інформаційних систем для ідентифікації істинних джерел затрат в організації, яку представляють за допомогою функціонального моделювання.

Вартісний аналіз представляє собою певну угоду щодо обліку витрат, які пов'язані з роботами, з метою визначити загальну вартість процесу. Вартісний аналіз базується на моделі робіт, тому що кількісна оцінка є неможливою без детального розуміння функціональності організації. Як правило, ABC застосовують для того, щоб зрозуміти походження вихідних затрат і полегшити вибір потрібної моделі робіт для процесу реорганізації діяльності організації (Business Process Reengineering, BRP). За допомогою вартісного аналізу можна вирішити наступні завдання, наприклад, визначення реальної вартості виробничого продукту, визначення реальної вартості підтримки клієнтів, ідентифікація робіт, які коштують найбільше (тобто, тих, які необхідно вдосконалювати першочергово), забезпечення менеджерів фінансовою міркою запропонованих змін і т.д.

Вартісний аналіз можна проводити тільки тоді, коли модель роботи є послідовною, тобто, побудованою згідно синтаксичних правил IDEF0, і коректною, а отже відображає відповідний процес, а також повною, тобто такою, що відображає всю область, що розглядається, і є стабільною, тобто пройшла цикл експертизи без змін. Все вище згадане передбачає, що створення моделі функціональних процесів має бути завершеним.

У ABC аналізі розрізняють такі поняття:

- Об'єкт затрат – причина, згідно якої робота виконується, як правило, це є основний вихід роботи. Вартість робіт є сумарною вартістю об'єктів затрат;
- Двигун затрат – є характеристикою входів і управління роботи, які впливають на тривалість і те, як виконується робота;

- Центри затрат – які представляють статті видатків.

Під час проведення вартісного аналізу у програмі BPwin початково задають одиниці вимірювання часу і грошей через меню Model – Model Properties – ABC Units. Якщо у списку відсутні відповідні позначення валюти, її можна доповнити. Часові одиниці вимірювання можна вибирати із широкого діапазону.

Наступним кроком аналізу є опис центрів затрат (Cost centers), який можна викликати за допомогою меню Model – Cost Center Editor. Кожному центру затрат необхідно дати детальний опис у вікні Definition.

Для задання вартості роботи (тобто для кожної роботи на діаграмі декомпозиції) необхідно у її контекстному меню вибрати команду Costs. У діалоговому вікні необхідно вказати частоту (Frequency) і тривалість (Duration) проведення цієї роботи у рамках загального процесу. Після цього у списку центрів затрат для кожного із центрів необхідно задати його вартість, тобто, у результаті задається вартість кожної роботи щодо кожної із статей видатків. Якщо у процесі визначення вартості виникає потреба у формуванні додаткових центрів затрат, можна викликати Cost Center Editor безпосередньо із наявного діалогового вікна.

Загальні затрати щодо даної роботи програма розраховує як суму щодо всіх центрів затрат. Під час обчислення затрат батьківської роботи спочатку обчислюється добуток затрат дочірньої роботи на частоту роботи (число разів, які робота виконується у рамках проведення батьківської роботи), а потім результати додаються. Якщо у всіх роботах моделі ввімкнено режим Compute from Decompositions, то такі обчислення автоматично проводяться у всій ієрархії робіт знизу вгору. Такий доволі простий спосіб підрахунку є адекватним, якщо роботи виконуються послідовно.

Якщо схема виконання робіт є складнішою, наприклад, роботи відбуваються альтернативно, то можна відмовитися від розрахунку і задати підсумкові величини для кожної роботи вручну (режим Override Decompositions). У цьому випадку результати роботи з нижніх рівнів будуть ігноруватися, і під час розрахунків на верхніх рівнях буде враховуватися сума, задана вручну.

Для проведення більш тонкого функціонально вартісного аналізу процесів можна скористатися спеціалізованими програмними продуктами, наприклад, EasyABC. Програма BPwin дає можливість експорту даних через меню File – Export. Зокрема, для передачі даних в EasyABC використовується експортування дерева вузлів у текстовий файл, який вже можна імпортувати у EasyABC. Зворотню процедуру можна реалізувати через меню File – Import – Costs.

Результати вартісного аналізу можуть суттєво впливати на черговість виконання робіт у функціональному процесі.

Результати вартісного аналізу наочно представляють у спеціальному звіті BPwin, налаштування якого можна провести у діалоговому вікні Tools – Reports – Activity Cost Report. Звіт дозволяє документувати ім'я, номер, визначення і вартість робіт як сумарно, так і в розрізі окремих центрів затрат. Результати вартісного аналізу відображаються безпосередньо і на діаграмах, у лівому нижньому куті роботи може відображатися вартість (за замовчуванням), або тривалість, або частота проведення роботи (налаштування через Model Properties - Display - ABC Data, ABC Units).

Аналіз за допомогою властивостей внесених користувачем.

ABC аналіз дозволяє оцінити вартісні і часові характеристики системи. Якщо вартісних показників є недостатньо, BPwin дає можливість внесення власних метрик, а саме властивостей визначених користувачем (User Defined Properties, UDP). UDP дозволяють провести додатковий аналіз, але без підсумовуючих підрахунків.

Описати UDP можна за допомогою User-Defined Property Editor (Model- UDP Definition Editor). У верхньому вікні діалогу вносять ім'я UDP, а у списку вибору Datatype можна описати його тип (18 різних типів). Серед них є управляючі команди і масиви, об'єднаних у категорії. Для внесення категорії необхідно задати ім'я категорії у вікні New Keyword і вибрати команду Add. Для присвоєння властивості до певної категорії необхідно вибрати UDP зі списку, а потім категорію із списку Keywords і вибрати кнопку Update. Одна категорія може об'єднувати декілька властивостей, в той же час одна властивість може входити у декілька категорій. Властивість типу List може містити масив попередньо визначених значень. Для визначення області значень UDP типу List необхідно задати значення властивості у вікні New Member і вибрати кнопку Add. Значення із списку можна редагувати.

Кожній роботі можна поставити у відповідність набір UDP. Для цього у контекстному меню роботи треба вибрати команду UDP. На закладці UDP Values можна задати значення UDP. Властивості List відображаються зі списком вибору, який заповнений попередньо значеннями. Властивості типу Command можуть мати у якості значення командний рядок, який виконується при виборі. Команда Filter використовується для задання фільтру за категоріями UDP. За замовчуванням у списку наявні властивості всіх категорій.

Хід виконання роботи:

Завдання 1. Побудова функціональної моделі роботи організації за зразком.

- 1) Після запуску програми All Fusion Process BPWin вказати назву моделі, а також вибрати методологію побудови моделі IDEF0. Для нової моделі викликати вікно Model Properties (властивості моделі) і заповнити відповідні вкладки:

Мета (Purpose), наприклад, „Моделювати поточні функціональні процеси організації” і вказати „точку огляду” Viewpoint, наприклад, „Експерт” (т.з. перспектива з якої спостерігається модельована система);

Визначення моделі і опис предметної області (Scope) (вкладка Definition), наприклад, „Навчальна модель, яка описує діяльність компанії і „Загальне управління функціонуванням організації”;

Статус даної розробки моделі (чорновик – draft, робочий варіант – working і т.д.) на вкладці Status;

Джерело інформації для побудови моделі – вкладка Source;

Доповнити ім'я проекту і моделі, авторів, часові рамки моделі AS-IS (так як є) і TO BE (як буде);

Задати вигляд нумерації в ієрархії робіт (наприклад, A0, A1 і т.п.) на вкладці Numbering;

Ознайомитись із вкладкою Display (відобразити), зокрема, для організації перегляду ICOM кодів.

Результат опису моделі, а також ряд інших звітів щодо компонентів моделі можна переглянути і роздрукувати за допомогою меню Tools – Model Report.

Для побудови контекстної діаграми моделі необхідно зобразити роботу (Activity), яка представляє собою поіменовані процеси, функції або завдання, що відбуваються протягом визначеного часу і мають результати, які добре розпізнаються. Необхідно також побудувати діаграми декомпозиції, які містять, т.з. дочірні роботи, що мають загального предка. Стрілки (Arrow) описують взаємодію робіт.

- 2) У вікні програми відображається блок роботи (Activity). Для побудови нульового листа діаграми (A0) задайте властивості блоку, зокрема, ім'я роботи, опис, її статус, шрифт напису, колір фону, шрифта, через діалогове вікно Activity Properties (активізується подвійним клацанням) або у контекстному вікні роботи (Рис. 1).
- 3) Для побудови ICOM стрілок до блоку A0 використовуйте граничні стрілки, тобто ті, які починаються або закінчуються біля границі діаграми (Рис. 1). Наприклад, для стрілки входу вибрати відповідну піктограму на панелі інструментів і активізувати чорне поле зліва і ліву частину блоку A0. Наступним етапом є задання імені і форматування стрілки через її контекстне меню (наприклад, „підрізати” довжину – команда „Trim”). Внутрішні стрілки на діаграмах декомпозиції зображаються аналогічно. Граничні стрілки на діаграмах декомпозиції появляються автоматично і потребують наступного зв'язування із потрібними блоками робіт.
- 4) Переглянути словник стрілок Arrow Dictionary (меню Dictionary). Імена внесених стрілок, занесених у словник можна також отримати у формі звіту Tools – Reports – Arrow Report.
- 5) Побудувати діаграму декомпозиції, використовуючи як підказку діаграму, зображену на рис.2. Для переходу до декомпонованого процесу і наверх використовуйте кнопки панелі інструментів Go to Child Diagram і Go to Parent Diagram. Під час декомпозиції слід вказати нотацію нової діаграми (у нашому випадку IDEF0) і вибрати кількість робіт у ній (у прикладі 3). Можна у процесі роботи над декомпонованою діаграмою додати необхідну кількість блоків робіт за допомогою панелі інструментів. Як правило, роботи на діаграмі декомпозиції розміщуються у порядку домінування зліва – направо. Візуальною ознакою наявності декомпозиції деякої роботи є відсутність невеликої діагональної риски у її лівому верхньому куті.
- 6) Для моделювання розгалуження або злиття функціональних процесів для дочірніх діаграм, як правило, використовують відповідні стрілки. Для того, щоб підписати такі стрілки необхідно виділити її частину до якої стосується відповідний напис і викликати контекстне меню, у якому задати назву та „прив'язку” командою Squiggle. Подібним чином підпишіть решту стрілок на діаграмі декомпозиції.
- 7) Тунелюйте стрілку Маркетингові матеріали (“не в батьківській діаграмі”) і зв'яжіть її на верхньому рівні (Change it to resolved rounded tunnel) за допомогою контекстного меню. Для стрілки зворотнього зв'язку по управлінню задайте властивість Extra Arrowhead.

- 8) Побудувати діаграму декомпозиції роботи A2 “Складання і тестування комп’ютерів”, за зразком зображеним на рис.3. Тунелюйте і зв’яжіть на верхньому рівні граничні стрілки “Компоненти” і “Персонал виробничого відділу” (“не в батьківській діаграмі”).

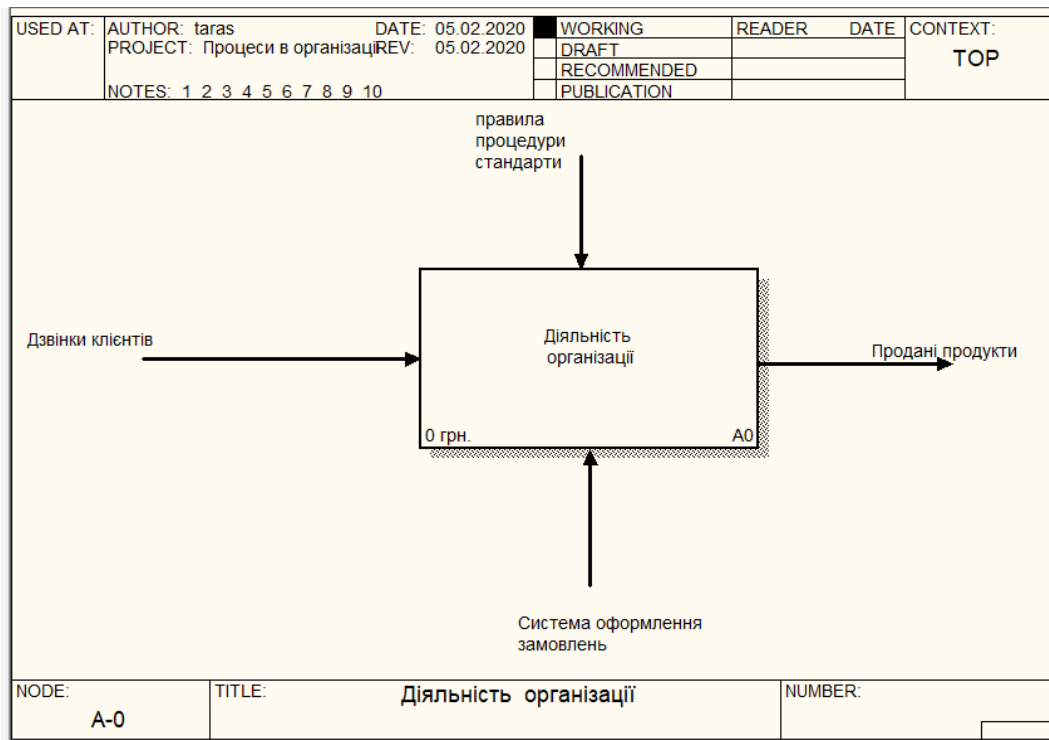


Рис.1. Приклад контекстної діаграми (блок A0).

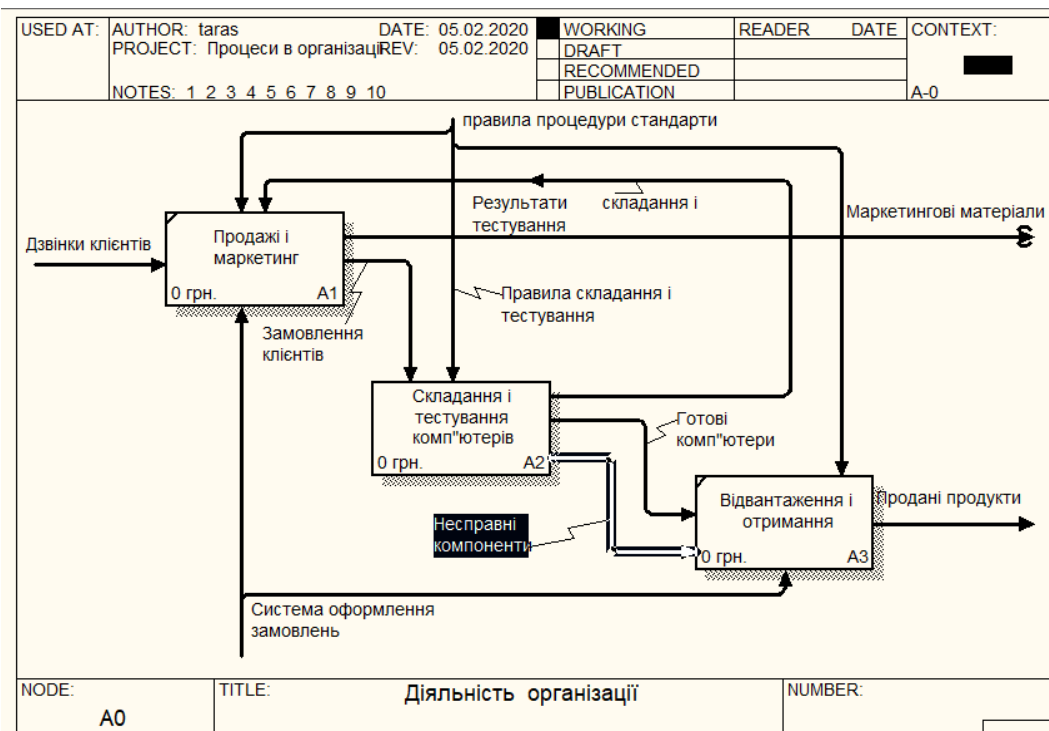


Рис.2. Приклад діаграми декомпозиції блоку A0.

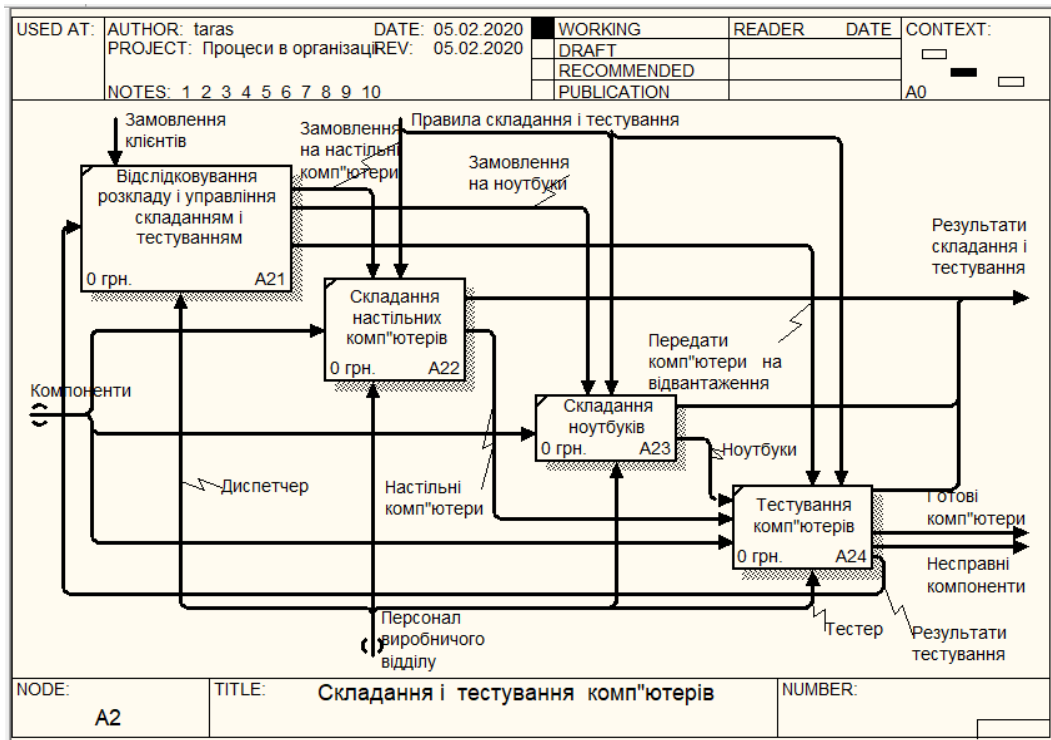
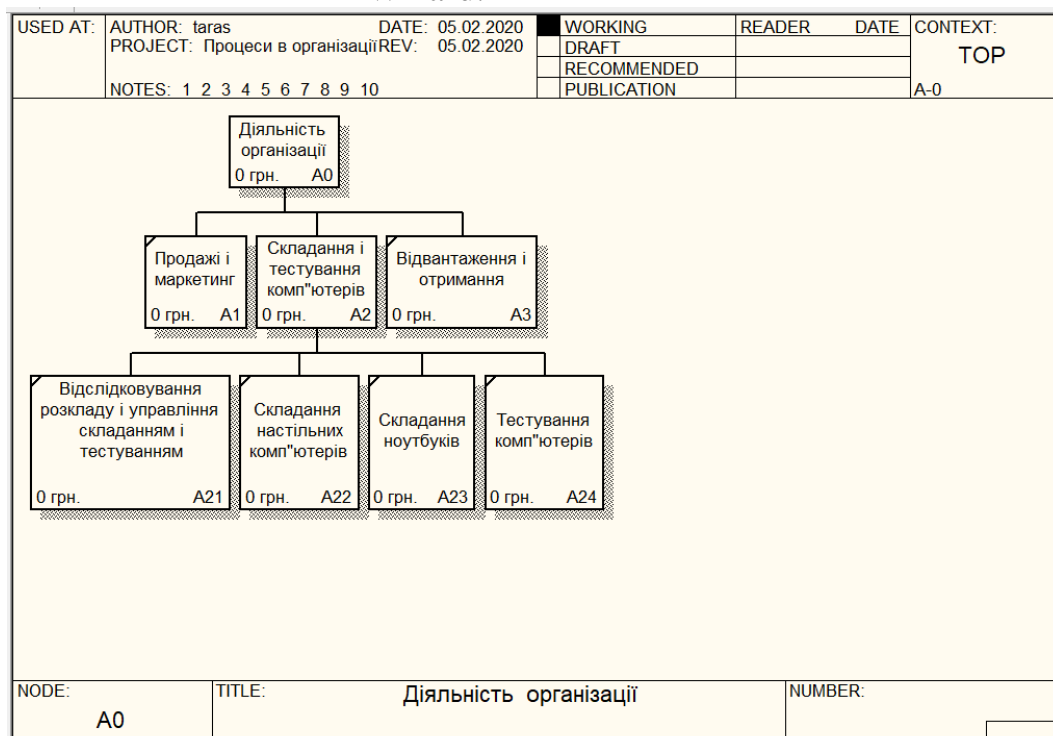


Рис.3. Приклад діаграми декомпозиції блоку A2.

- 9) Ознайомтесь і проаналізуйте структуру каркасу для побудованих діаграм. Каркас містить заголовок і нижню частину. Заголовок, як правило, використовують для відслідковування діаграми в процесі моделювання. Нижня частина застосовується для ідентифікації і позиціонування в ієрархії діаграм. Дані відображені у каркасі можуть бути вибрані за допомогою меню Diagram – Diagram Properties.
- 10) Переглянути структуру моделі через Model Explorer (меню View).
- 11) Побудуйте діаграму дерева вузлів Node Tree (меню Diagram – Add Node Tree) (Рис.4). Задайте назву дерева, вкажіть кореневу діаграму дерева і кількість рівнів відображення, а також різні представлення дерева через діалогові вікна Node Tree Wizard.



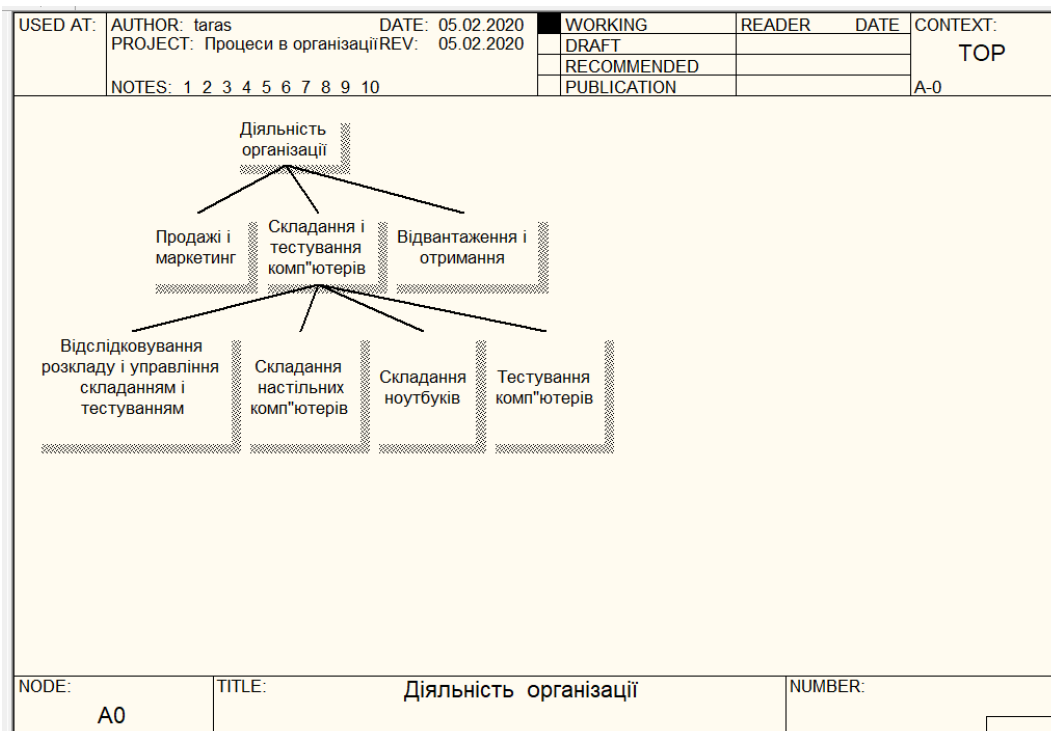


Рис.4. Приклади представлення дерева вузлів функціональної моделі:
 а) ортогональні лінії гілок;
 б) діагональні лінії гілок.

- 12) Побудуйте окремий альтернативний варіант частини моделі з використанням діаграм тільки для експозиції (FEO – for exposition only) через меню Diagram – Add FEO Diagram. FEO діаграми можуть використовуватися в моделі для ілюстрації інших точок зору (Viewpoint), для відображення окремих деталей, які не підтримуються явно синтаксисом IDEF0. Такі діаграми є просто картинками – копіями стандартних діаграм і не включені до аналізу синтаксису. Наприклад, розглянемо взаємодію роботи “Складання і тестування комп’ютерів” із іншими роботами. Для цього створимо FEO діаграму, яка міститиме тільки стрілки цієї роботи. Для цього у діалоговому вікні задайте ім’я діаграми, виберіть тип зображення діаграми декомпозиції, а також діаграму – джерело для створення діаграми FEO, наприклад, A2 “Складання і тестування комп’ютерів”. Видаліть деякі стрілки на діаграмі FEO (Рис. 5).

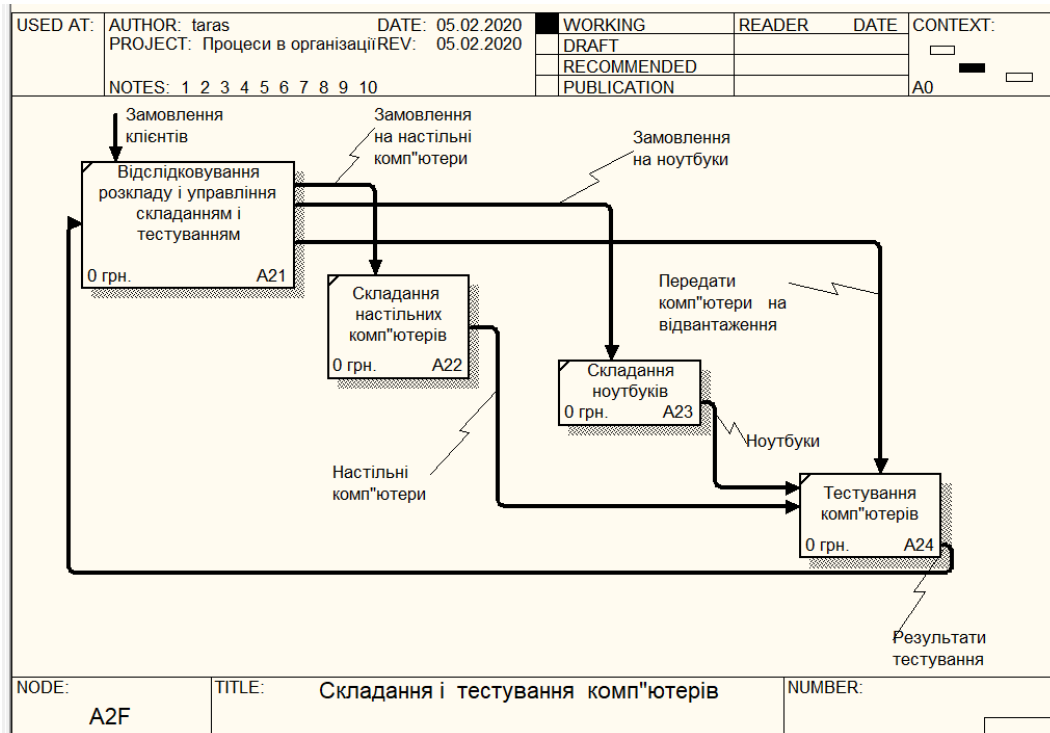


Рис. 5. Приклад діаграми FEO A2F, яка дозволяє сконцентруватись на певній частині діаграми декомпозиції, без дотримання повного синтаксису IDEF0.

- 13) Ознайомтесь із звітами щодо побудованої моделі (меню Tools – Reports), особливо із Model Consistency Report (звіт про цілісність моделі), який може містити список синтаксичних помилок моделі. Приклад вікна перегляду звіту наведено на Рис. 6.

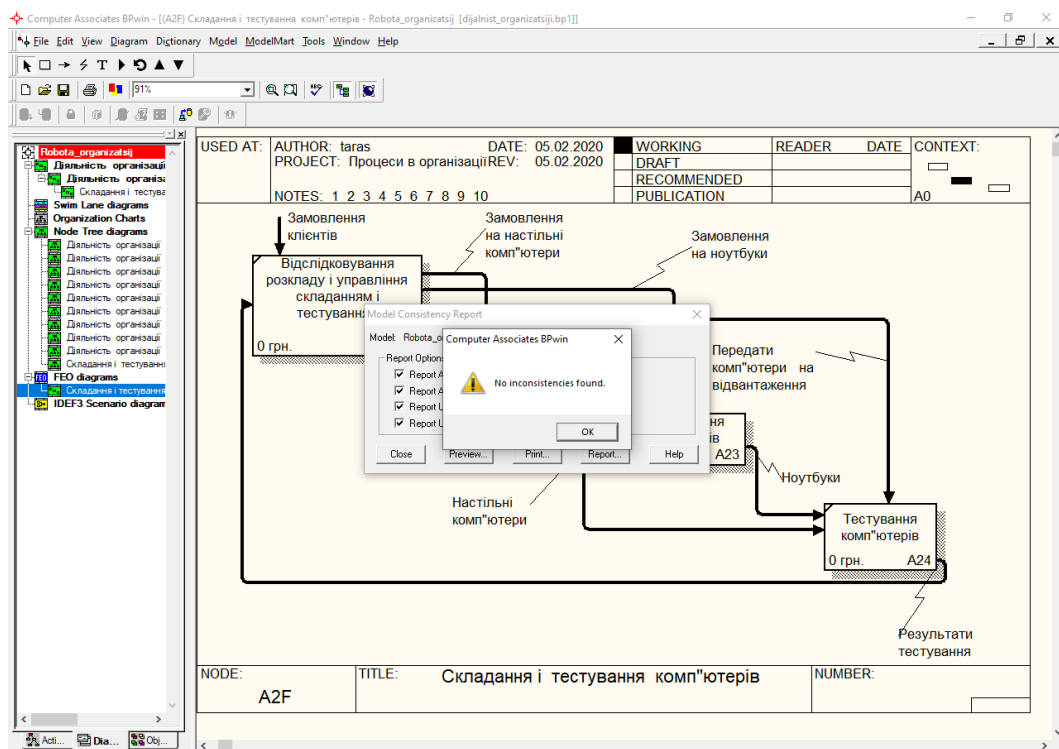


Рис 6. Приклад діалогу під час аналізу повноти функціональної моделі.

Завдання 2. Робота з організації злиття і розщеплення моделей для підготовки проекту до колективної роботи.

- 1) Проведіть розщеплення в копії побудованої моделі із Завдання 1, використовуючи роботу “Складання і тестування комп’ютерів”, в якості роботи від якої буде відщеплена окрема модель. Так процедура може бути виконана, оскільки дана робота є декомпонованою, зокрема, у нашому випадку на 4 дочірні роботи. В контекстному меню роботи виберіть команду Split Model і в діалоговому вікні Split Options вкажіть ім’я моделі, яку відщеплюєте, наприклад, Виробничий процес, а також опцію Enable Merge/Overwrite Option. У старій моделі стрілка виклику, що появилась, буде мати ім’я Виробничий процес, а робота “Складання і тестування комп’ютерів” стане недекомпонованою (Рис. 7).

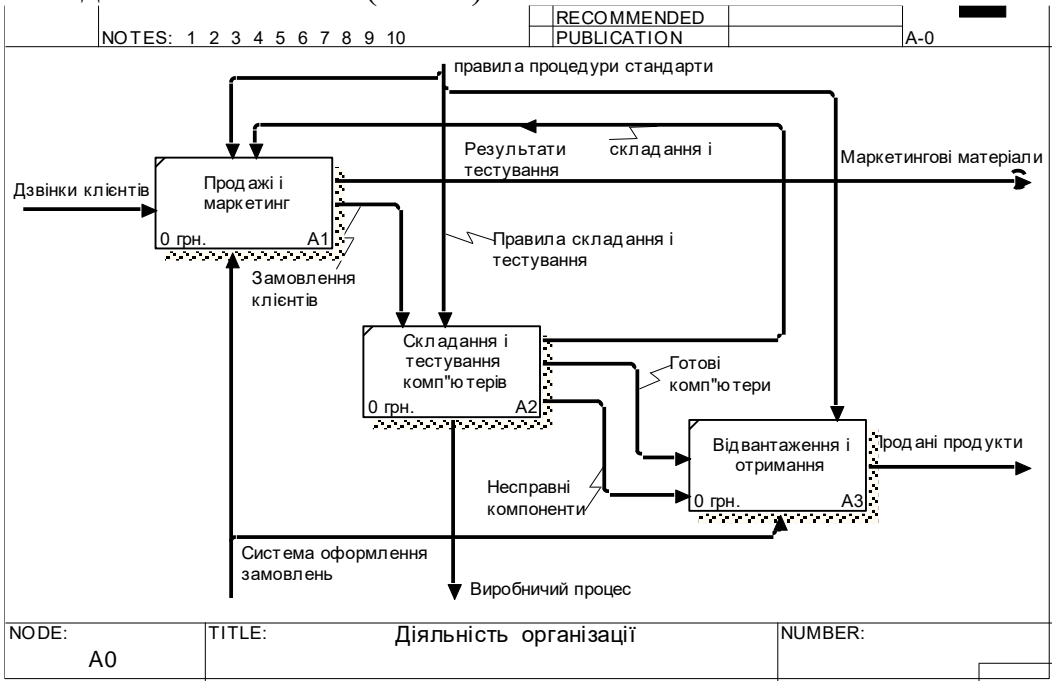


Рис. 7. Приклад діаграми вихідної моделі після проведення відщеплення.

На рис. 8 зображено контекстну діаграму нової моделі Виробничий процес з іменем “Складання і тестування комп’ютерів”.

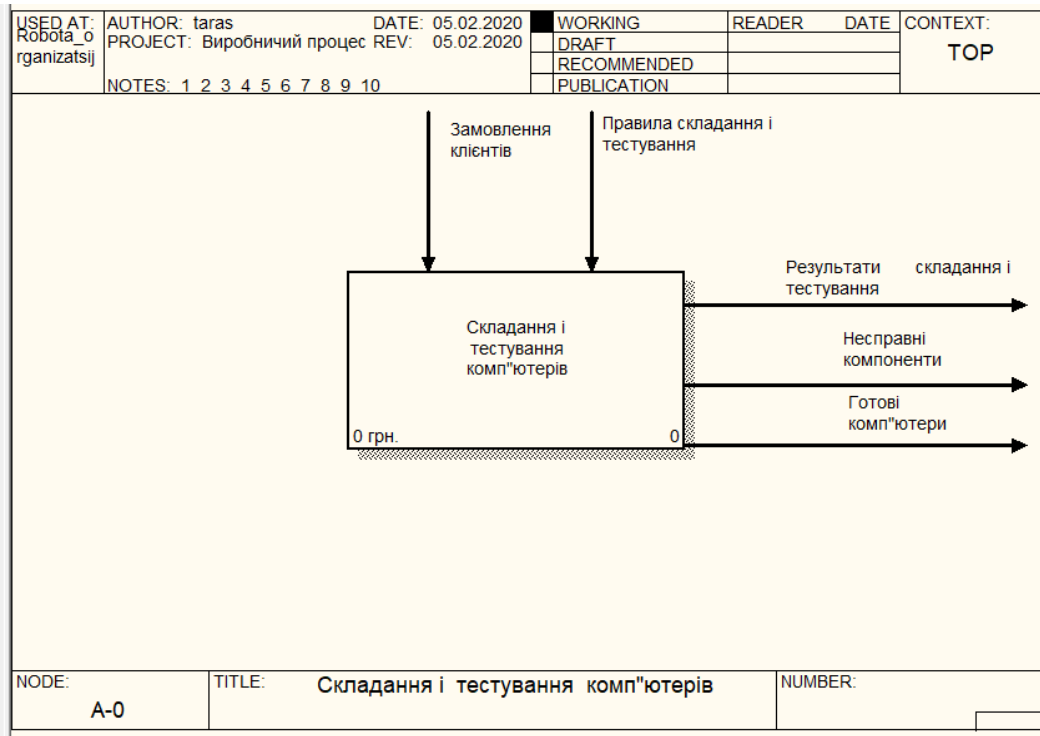


Рис. 8. Контекстна діаграма нової моделі Виробничий процес.

- 1) Задайте властивості нової моделі. Її можна зберегти під новим іменем. Оскільки в старій моделі робота з якої проведено відщеплення була декомпонованою, то у новій моделі можна побачити, окрім, контекстної моделі, також відповідну діаграму декомпозиції (Рис. 9).

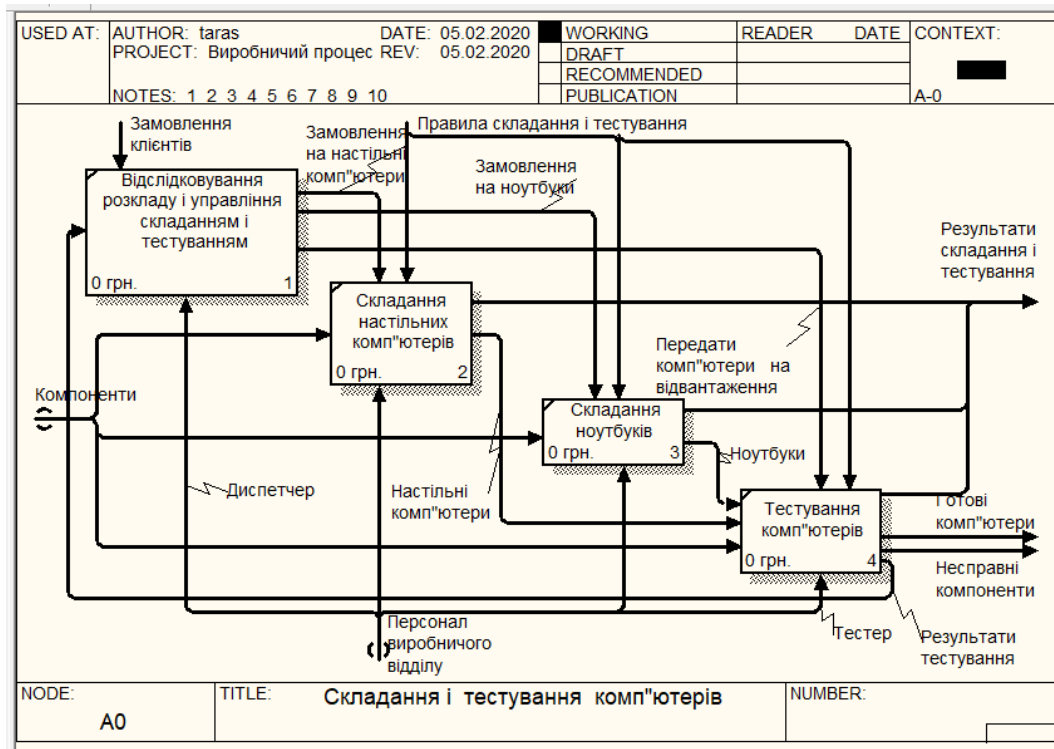


Рис. 9. Діаграма декомпозиції у новій моделі Виробничий процес.

- 2) Аналогічно проведіть відщеплення нової моделі від копії вихідної моделі, використовуючи для цього роботу "Відвантаження і отримання" (Рис. 10).

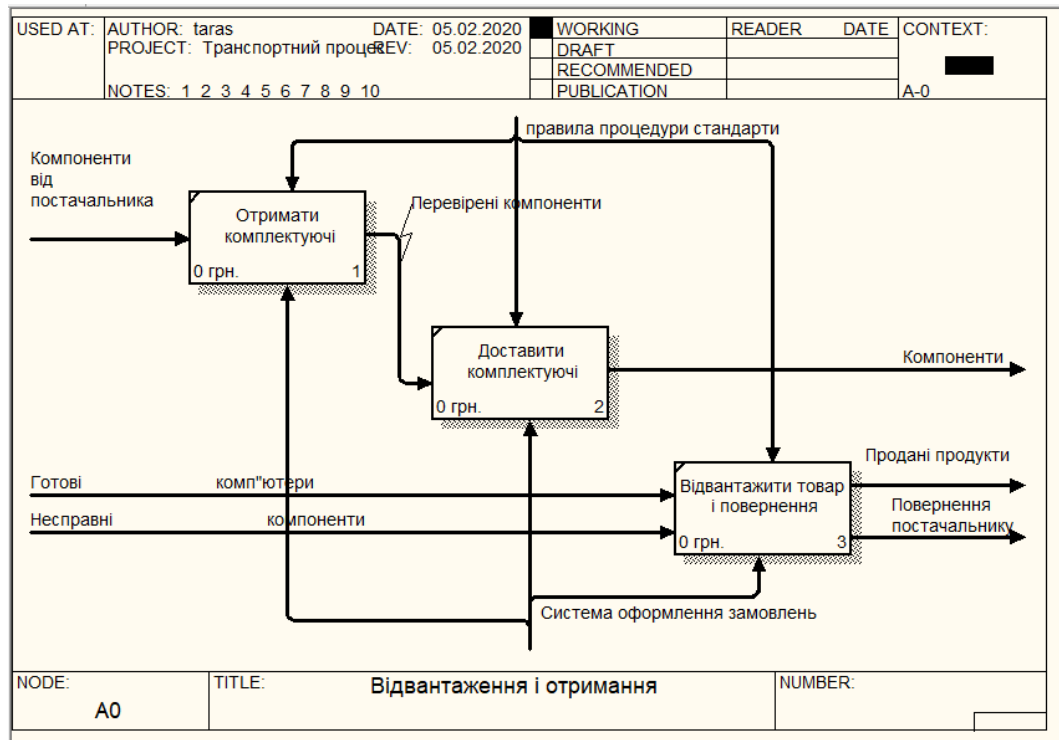


Рис. 12. Діаграма декомпозиції в моделі Транспортний процес.

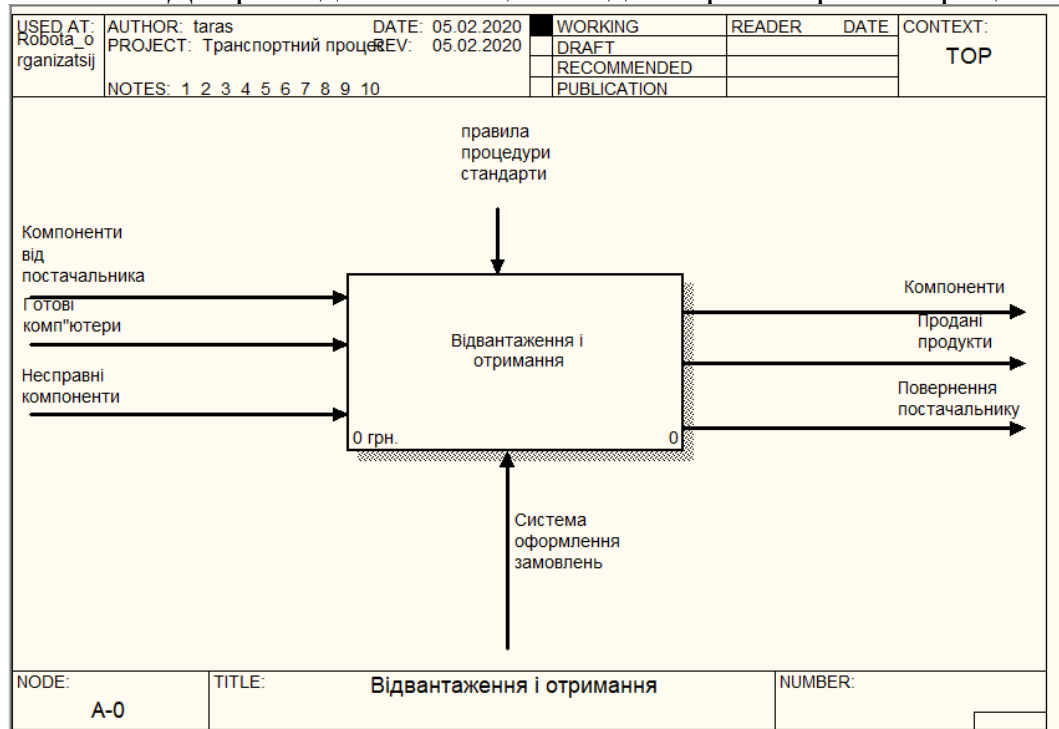


Рис. 13. Уточнена контекстна діаграма моделі Транспортний процес.

- 4) Проведіть процедуру злиття моделей із використанням моделей Процеси в організації (вихідна модель на Рис. 10) і Транспортний процес (Рис. 13). Для цього обидві моделі мають бути відкриті в BPwin. Виберіть на діаграмі вихідної моделі роботу “Відвантаження і отримання” і в контекстному меню команду Merge Model. В діалоговому вікні виберіть опцію Cut/Paste Entire Dictionaries. Для роботи “Відвантаження і отримання” зникає стрілка виклику і вона стає декомпонованою роботою. У результуючій діаграмі тунелюйте і зв’яжіть стрілки як зображено на Рис.14, зокрема, стрілка Компоненти, яка

виходить із роботи “Відвантаження і отримання” тепер стає внутрішньою стрілкою діаграми.

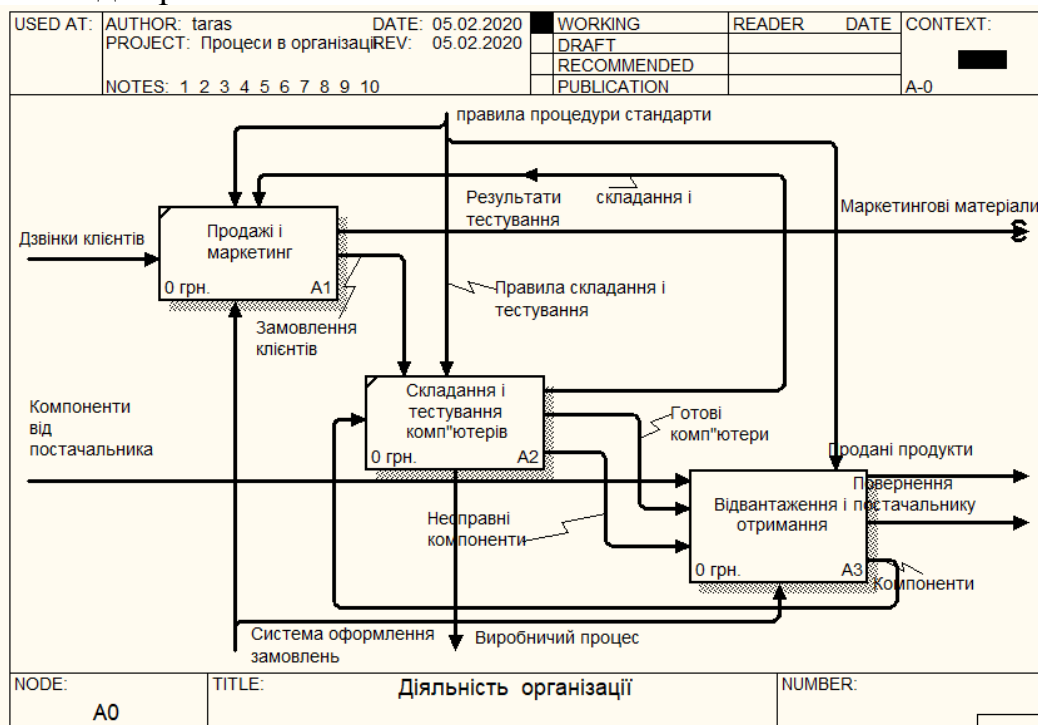


Рис. 14. Результуюча модель після виконання процедури злиття.

- 5) Проаналізуйте можливість використання готових моделей або їх компонентів для побудови нових моделей. Для цього проведіть копіювання робіт між моделями. Створіть нову модель, наприклад, з іменем Тест. Декомпонуйте контекстну модель в новій моделі, не вносячи імена робіт. Запустіть Model Explorer (меню View – Model Explorer) і перейдіть на вкладку Activity. В режимі drag and drop перетягніть будь-яку із робіт, які є у вже збудованій моделі на діаграму декомпозиції моделі Тест у вікні Model Explorer. У діалоговому вікні Continue with Merge виберіть опцію Cut/Paste Entire Dictionaries і скопіюйте роботу із вихідної діаграми на нову діаграму моделі Тест. Таким же чином можна реалізувати перенос робіт в межах однієї і тієї ж моделі.

Провести роботу із оцінювання функціональних моделей відповідно методик вартісного оцінювання ABC і UDP.

Завдання 3. ABC аналіз моделі.

1. Для виконання завдання використайте готову синтаксично коректну модель Робота організації із Завдання 1.
2. Проведіть ABC аналіз процесів. Встановіть у діалоговому вікні Model Properties – ABC Units одиниці вимірювання грошей і часу (Рис.1).
3. Внесіть назву і визначення для центрів затрат, використовуючи команду Dictionary – Cost Center. Для підтвердження вводу даних використовуйте клавішу табуляції (Рис.2).
4. Для відображення (не відображення) на позначенні роботи її вартості у діалоговому вікні Model Properties – Display виберіть опцію ABC Data.

5. Встановіть вартості робіт через контекстне меню робіт Costs на діаграмі декомпозиції блоку A2 згідно Таблиці 1. Перегляньте на діаграмах результат внесення даних.
6. Згенеруйте звіт щодо вартості робіт Activity Cost Report через меню Tools – Reports. Налаштуйте вигляд звіту у діалоговому вікні, зокрема, задайте опції Activity Name, Activity Costs, Cost Center Name, Cost Center Cost. У якості формату звіту виберіть DDE Table (Рис. 3).
7. Попередньо перегляньте вміст звіту у випадку стандартних звітів щодо центрів затрат і вартостей робіт на рівнях декомпозиції і контекстної діаграми моделі та згенеруйте і збережіть отриманий звіт у файл MS Word (Рис. 4).

Таблиця 1

Вартості робіт на діаграмі декомпозиції блоку A2

Ім'я роботи Activity Name	Центр затрат Cost center	Вартість центру затрат, грн. Cost Center Cost	Частота Frequency	Тривалість, днів Duration
Відслідковування розкладу і управління складанням і тестуванням	Управління	50,00	1,00	1,00
Складання настільних комп'ютерів	Робоча сила	10,00	12,00	1,00
	Компоненти	1600,00		
Складання ноутбуків	Робоча сила	14,00	20,00	1,00
	Компоненти	2800,00		
Тестування комп'ютерів	Робоча сила	6,00	32,00	1,00

Model Properties

General Purpose Definition Source Status Numbering Display
Layout ABC Units Page Setup Header/Footer Shapes Draw Style

Model Name:
Robota_organizatsij

Cost

Currency description: грн. Symbol placement: 1 грн.

Symbol: грн. Number of decimals in diagrams: 0 Number of decimals in reports: 2

Time

Time Unit: Days Decimals in frequency values: 2 Decimals in duration values: 2

OK Отмена Применить Справка

Рис. 1. Налаштовування одиниць вимірювання грошей і часу.

Cost Center Dictionary	
Name	Definition
Upravlinnja	Zatraty na upravlinnja
Robocha syla	Zatraty na oplatu pratsi
Komponenty	Zatraty na zakupku komponentiv

Рис. 2. Назви і визначення центрів затрат.

Activity Based Costing Report

Standard reports: Activity Cost / Cost Center (landscape) [Update] [New] [Delete]

Model: Robota_organizatsij

Start from activity: A2: Складання і тестування комп'ютерів

Activity Options

1 ☒ Activity name
☐ Activity number
☐ Definition
2 ☒ Activity costs

Cost Center Options

3 ☒ Cost center name
☐ Cost center definition
4 ☒ Cost center costs

Activity Ordering

☐ Alphabetical
☒ Hierarchical
☐ Breadth First

Time Period Options

☐ Activity duration
☐ Activity frequency

Report Format

☐ Labeled
☐ Fixed column
☐ Tab delimited
☐ Comma delimited
☒ DDE table
☐ RPTwin

Multi-Valued Format

☐ Repeating group
☐ Filled
☒ Header ☒ Merge

☒ Remove special char
☒ Column headings

ABC Options

☐ Calculate all non-leaf node costs
☐ Calculate all cost center totals

Format Options

☐ Report cost centers only
☒ Report cost centers within activity
☐ Report activities within cost centers

[Close] [Preview...] [Print...] [Report...] [Help]

Рис. 3. Вікно налаштування параметрів звіту ABC.

a)

Activity Name	Activity Cost (грн.)	Cost Center	Cost Center Cost (грн.)
Складання і тестування комп'ютерів	75 842,00	Komponenty	75 200,00
		Robocha syla	592,00
		Upravlinnja	50,00
Відслідковування розкладу і управління складанням і тестуванням	50,00	Upravlinnja	50,00
Складання настільних комп'ютерів	1 610,00	Komponenty	1 600,00
		Robocha syla	10,00
Складання ноутбуків	2 814,00	Komponenty	2 800,00
		Robocha syla	14,00
Тестування комп'ютерів	6,00	Robocha syla	6,00

б)

Cost Center	Cost Center Definition	Cost Center Cost (грн.)
Komponenty	Zatraty na zakupku komponentiv	75 200,00
Robocha syla	Zatraty na oplatu pratsi	592,00
Upravlinnja	Zatraty na upravlinnja	50,00

в)

Activity Name	Activity Number	Activity Cost (грн.)
---------------	-----------------	----------------------

Activity Name	Activity Number	Activity Cost (грн.)
Діяльність організації	A0	75 842,00
Продажі і маркетинг	A1	0,00
Складання і тестування комп'ютерів	A2	75 842,00
Відслідковування розкладу і управління складанням і тестуванням	A21	50,00
Складання настільних комп'ютерів	A22	1 610,00
Складання ноутбуків	A23	2 814,00
Тестування комп'ютерів	A24	6,00
Відвантаження і отримання	A3	0,00

Рис. 4. Приклади згенерованих звітів, збережених у форматі Word, внаслідок встановлення DDE зв'язків :

- а) Activity Cost/Cost Center (landscape), який починається з рівня роботи A2;
 б) за центрами затрат Cost Center; в) за іменами робіт Activity Cost, які починаються з рівня роботи A0.

Завдання 4. UDP аналіз.

1. Проведіть аналіз процесів із використанням властивостей, визначених користувачем UDP.
2. Для цього в меню Dictionary – UDP Keywords і задайте ключові слова: витрата ресурсів, документація, інформаційна система (Рис.4).

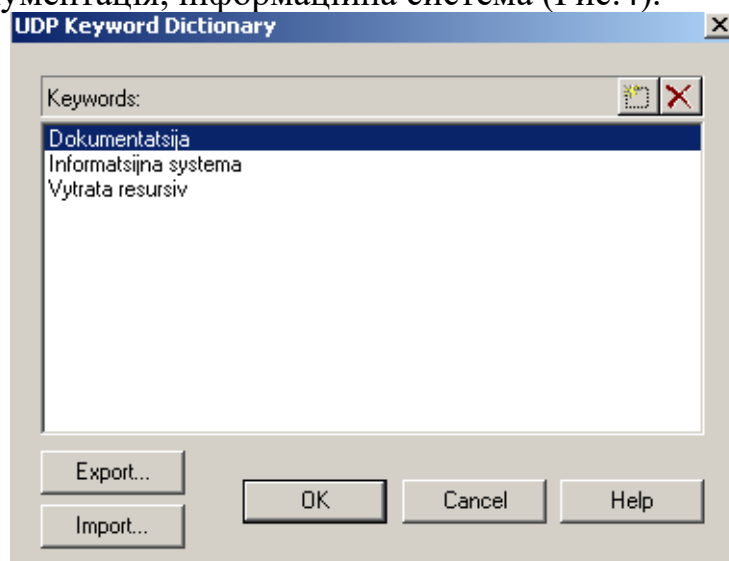


Рис. 4. Словник ключових слів UDP.

3. Командою Dictionary – UDP відкрийте словник UDP і задайте відповідні значення з Таблиці 2. Для під'єднання до UDP ключового слова перейдіть до відповідного поля Keyword. Для UDP типу List необхідно в полі Value задати список значень. Для підтвердження вводу даних використовуйте табуляцію. На рис. 5 наведено словник UDP після вводу даних Таблиці 2.

Таблиця 2

Найменування UDP (Name)	Тип (UDP Datatype)	Значення (Value)	Ключове слово (Keyword)
Прикладні програми	Text List (Multiple Selection)	Модуль оформлення замовлень Модуль створення і контролю розкладу виконання робіт Модуль врахування комплектуючих і обладнання Модуль процедур складання і пошуку несправностей	Інформаційна система
Додаткова документація	Command List	Winword.exe sample1.doc Winword.exe sample2.doc Powerpnt.exe sample3.ppt	Документація
Історія змін	Paragraph Text		Документація
Забруднення оточуючого середовища	Text List (Single Selection)	Дуже високе Високе Середнє Низьке	
Витрата електроенергії	Real Number		Витрата ресурсів

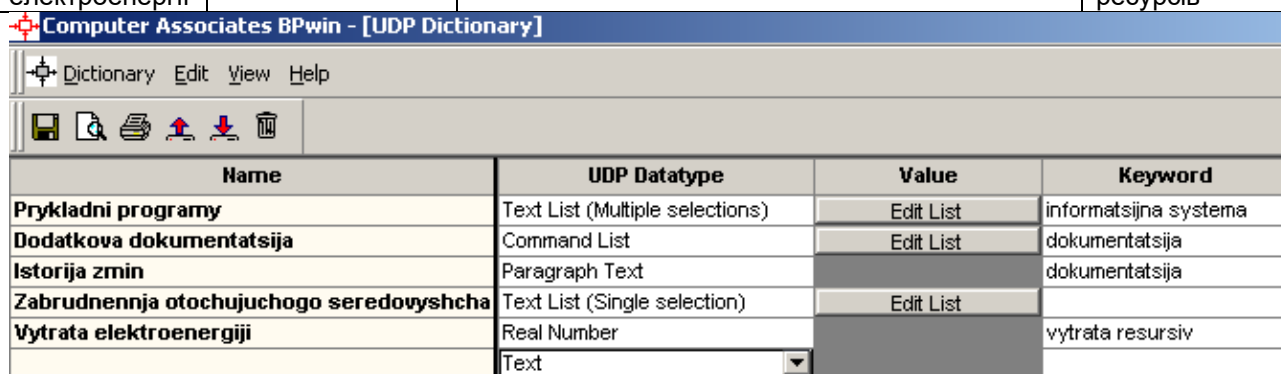


Рис. 5. Словник UDP.

4. Для призначення властивостей UDP певній роботі необхідно у її контекстному меню вибрати UDP. Введіть значення UDP для робіт на діаграмі декомпозиції блоку A2 згідно із Таблицею 3. Приклад, для роботи A21 наведено на рис. 6.

Таблиця 3.

Activity Name	Додаткова документація	Прикладні програми	Історія змін	Витрата електроенергії	Забруднення оточуючого середовища
Відслідковування розкладу і управління складанням і тестуванням	Winword.exe sample2.doc	Модуль створення і контролю розкладу виконання робіт	Історія змін специфікацій	10,00	Низьке
Складання настільних комп'ютерів		Модуль врахування комплектуючих і обладнання Модуль процедур складання і пошуку несправностей		20,00	Середнє
Складання ноутбуків		Модуль врахування комплектуючих і обладнання Модуль процедур складання і пошуку несправностей		25,00	Середнє
Тестування комп'ютерів		Модуль врахування комплектуючих і обладнання Модуль процедур складання і пошуку несправностей		40,00	Середнє

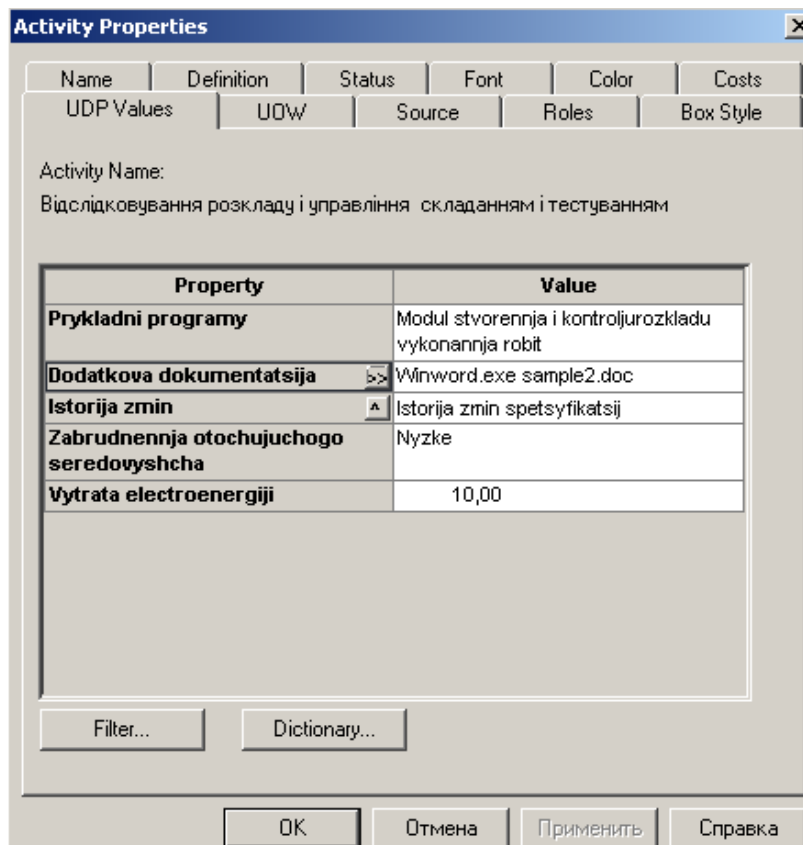


Рис. 6. Приклад внесення UDP властивостей для роботи A21.

Після внесення UDP типу Command або Command List вибір кнопки у вікні Activity Properties приводить до запуску відповідної прикладної програми і документу. Для цього створіть попередньо у тому самому каталозі, що і модель, наприклад, файл sample2.doc.

5. Задайте можливість фільтрації за ключовими словами для діалогового вікна Activity Properties кнопкою Filter і у вікні діалогу відключіть деякі із ключових слів. Після цього у діалоговому вікні Activity Properties не будуть відображатися UDP пов'язані із відповідними ключовими словами.
6. Створіть звіт за властивостями за допомогою меню Tools – Report – Diagram Object Report. Виберіть опції звіту почати з рівня A2, число рівнів 2, UDP – витрата електроенергії, DDE Table. Приклад згенерованого UDP звіту у форматі Word наведено на рис.7.

Activity Name	Vytrata electroenergiji
Складання і тестування комп'ютерів	
Відслідковування розкладу і управління складанням і тестуванням	10,00
Складання настільних комп'ютерів	20,00
Складання ноутбуків	25,00
Тестування комп'ютерів	40,00

Рис. 7. Приклад згенерованого звіту, збереженого у форматі Word, внаслідок встановлення DDE зв'язків.

Завдання 5. Проектування функціональності програмних систем в IDEF0.

Таблиця розв'язків квадратного рівняння

Номер розв'язку - n	A	B	C	D	Вигляд розв'язків
1	0	0	0		s
2	0	0	≠0		s
3	0	≠0			x
4	≠0			≥0	x1, x2
5	≠0			<0	s

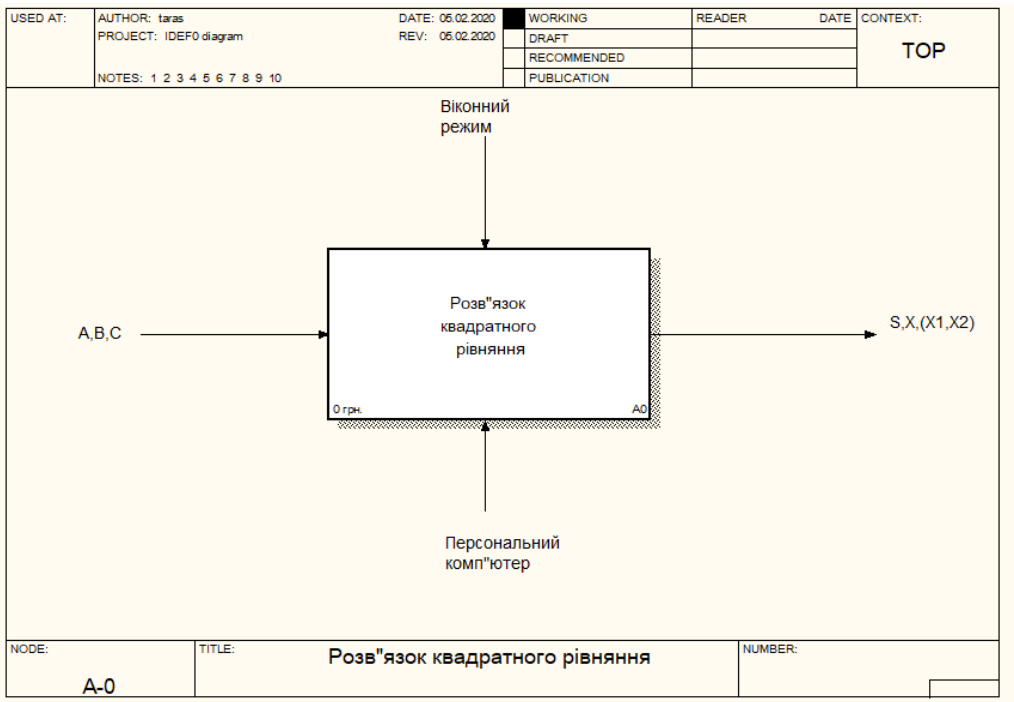


Рис.1. Контекстна діаграма.

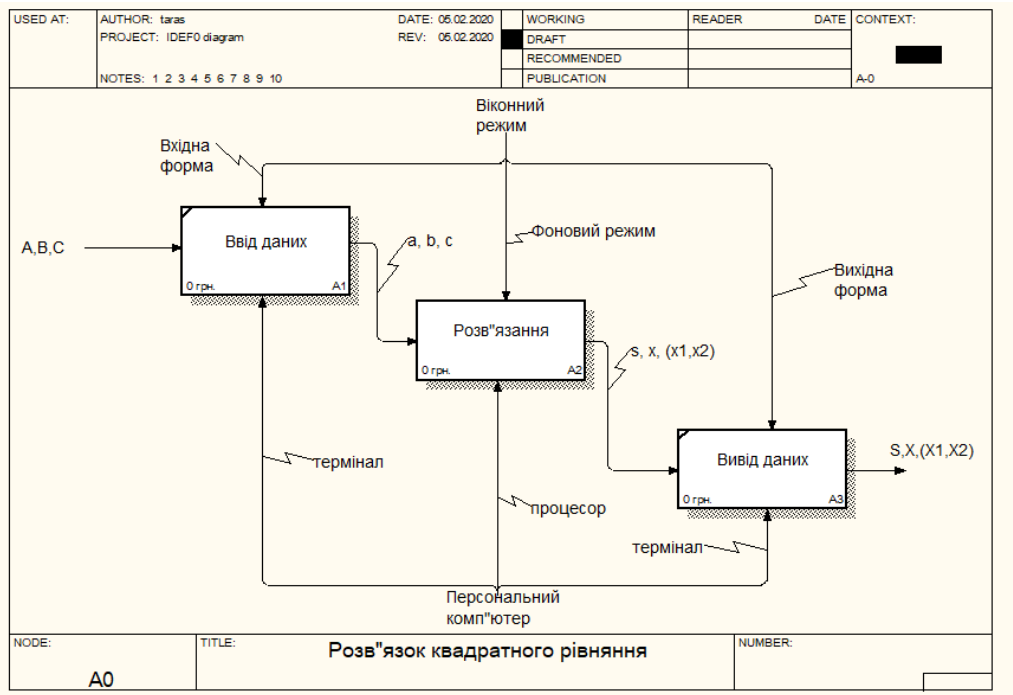


Рис.2. Діаграма декомпозиції блоку A0.

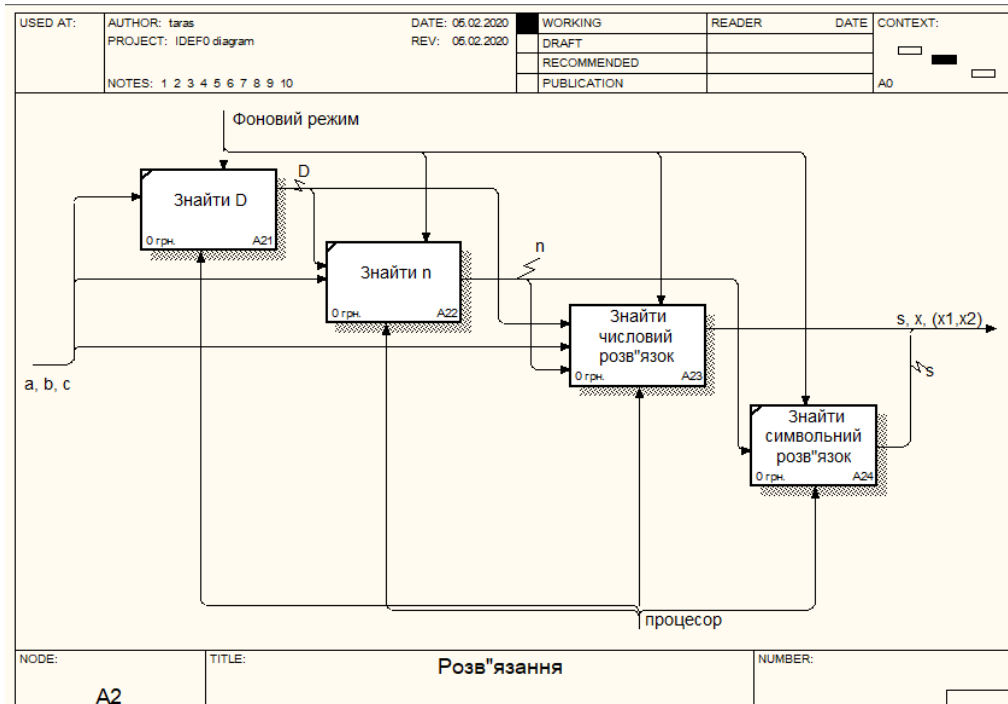


Рис.3. Діаграма декомпозиції блоку A2.

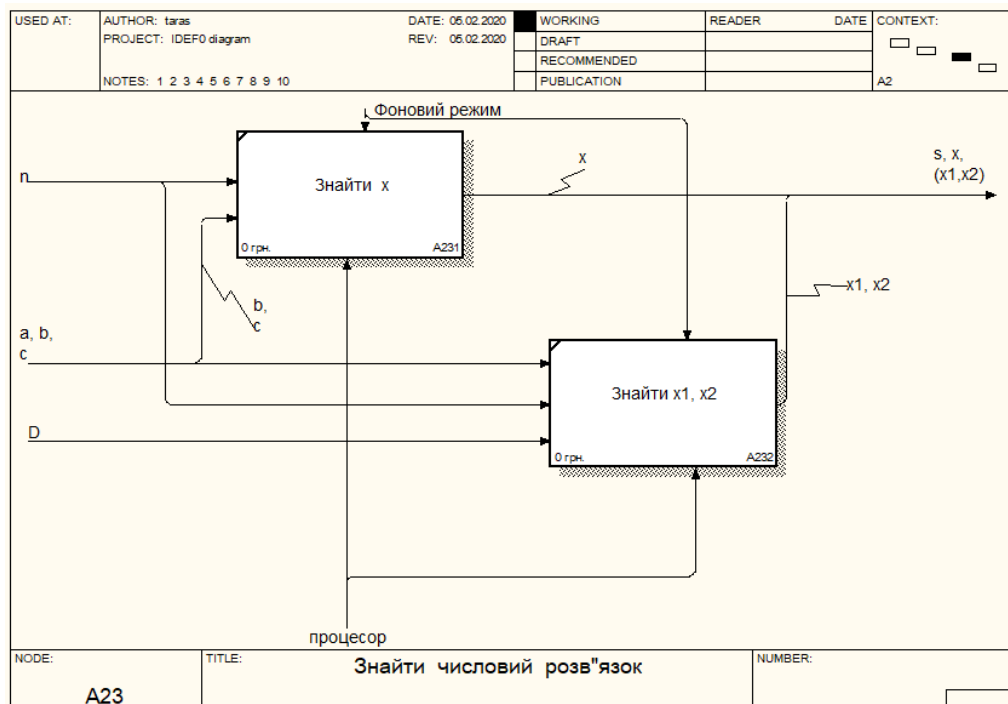


Рис.4. Діаграма декомпозиції блоку A23.

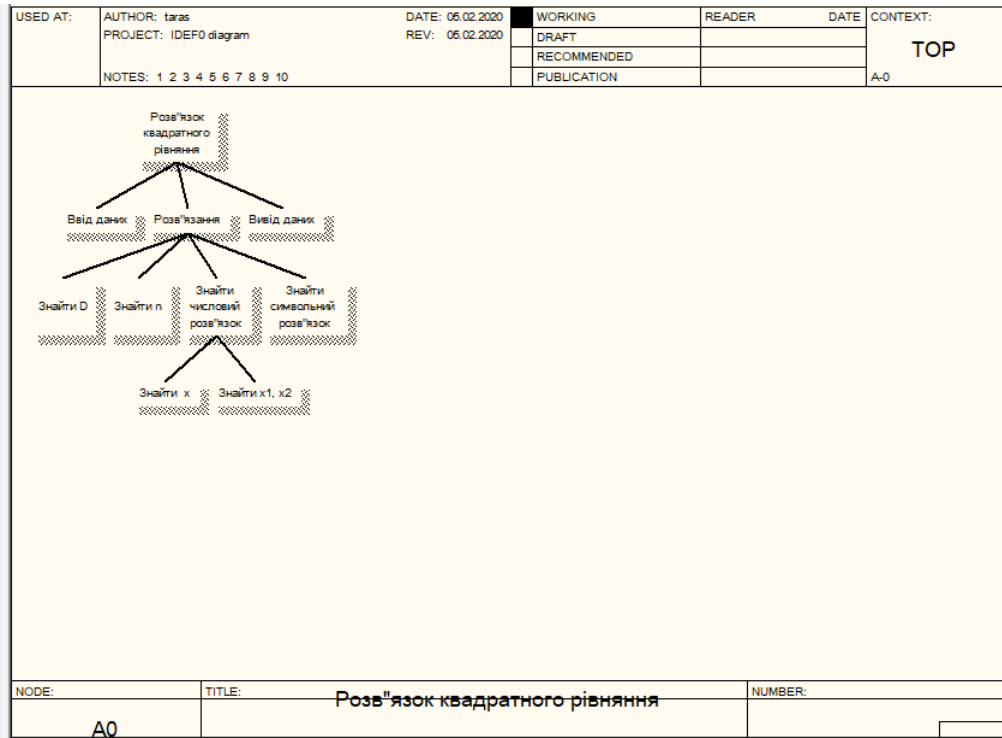


Рис.5. Дерево моделі.

Завдання 6. Побудова функціональної моделі інформаційної системи оцінювання успішності студентів із використанням технології IDEF0.

- 1) За допомогою All Fusion Process Modeler (BPwin) побудувати функціональну модель інформаційної системи оцінювання успішності студентів згідно із наданими критеріями оцінювання студентів факультету електроніки та комп'ютерних технологій.

Деякі рекомендації до побудови моделі:

Кількість рівнів декомпозиції не менше трьох.

Загальний процес – оцінювання навчального процесу студента, ІСОМ мітки – вхідні дані у вигляді оцінок згідно критеріїв оцінювання, журналів, критерії оцінювання як управління процесом, механізм – програмна реалізація системи, вихідні дані – у вигляді оцінок у заліковій/екзаменаційній відомості, оцінок в заліковій книжці (розглянути представлення у різних системах оцінювання: ECTS, національна шкала, шкала Університету).

Діаграми декомпозиції можуть містити наступні функціональні процеси: відмітки про відвідування лекцій, практичних і лабораторних занять, оцінювання за результатами проміжної атестації, оцінювання рефератів, оцінювання виконання лабораторних (практичних) завдань і оформлення звітів. Також діаграма може містити процеси оцінювання за результатами підсумкової атестації, сумарного оцінювання за курс для предметів, які закінчуються здачею заліку, підсумкове оцінювання курсу за результатами семестру і екзамену, а також інші процеси залежно від структури навчального плану курсу та критеріїв оцінювання, зокрема, забезпечити зворотні зв'язки по входу для оцінювання за талонами.

Згідно із нотацією IDEF0 не варто переобтяжувати певний рівень моделі кількістю процесів більше 6 (максимально 8). В разі потреби деталізації робіт краще звернутись до їх глибшої декомпозиції.

Наприклад, деталізацію процесу підсумкового оцінювання із виходом оцінок як у міжнародній шкалі, так і національній шкалі можна винести на дочірню діаграму, подібно, як і процесів підсумкового оцінювання за семестр.

Критерії оцінювання успішності студентів факультету електроніки та комп'ютерних технологій

Для предметів, вивчення яких закінчується здачею іспиту:

Підсумкова кількість балів виставляється по наступним критеріям:

Іспит – максимальна кількість балів 50.

Контроль успішності протягом семестру – максимальна кількість балів 50.

Іспит проводиться в письмовій формі з наступною усною співбесідою. Додаткові питання обов'язково записуються в екзаменаційну роботу студента.

При оцінюванні знань студента протягом семестру можливі наступні ситуації.

Предмет включає в себе:

1. **Лекції.** Можливі наступні форми контролю знань:

- проміжна атестація, яка може проводитись у формі колоквіуму або контрольної роботи,
- підсумкова атестація, яка може проводитись у формі колоквіуму або контрольної роботи,
- написання реферату по програмі курсу.

2. **Лекції і практичні.** Можливі наступні форми контролю знань:

- проміжна атестація, яка може проводитись у формі колоквіуму або контрольної роботи,
- підсумкова атестація, яка може проводитись у формі колоквіуму або контрольної роботи,
- написання реферату по програмі курсу.
- контроль успішності на практичних заняттях, який може включати в себе контроль на кожному практичному занятті, виконання домашніх завдань, написання контрольних робіт

3. **Лекції та лабораторні заняття.** Можливі наступні форми контролю знань:

- проміжна атестація, яка може проводитись у формі колоквіуму або контрольної роботи,
- підсумкова атестація, яка може проводитись у формі колоквіуму або контрольної роботи,
- написання реферату по програмі курсу.
- контроль успішності на лабораторних заняттях, який включає в себе бали за виконання кожної лабораторної роботи.

Для предметів, вивчення яких закінчується здачею заліку:

Залік виставляється по підсумковій кількості балів набраних студентом протягом семестру.

Максимальна кількість балів 100.

При оцінюванні знань студента протягом семестру можливі наступні ситуації.

Предмет включає в себе:

1. **Лекції.** Можливі наступні форми контролю знань:

- проміжна атестація, яка може проводитись у формі колоквіуму або контрольної роботи,
- підсумкова атестація, яка може проводитись у формі колоквіуму або контрольної роботи,
- написання реферату по програмі курсу.

2. **Лекції та лабораторні заняття.** Можливі наступні форми контролю знань:

- проміжна атестація, яка може проводитись у формі колоквіуму або контрольної роботи,
- підсумкова атестація, яка може проводитись у формі колоквіуму або контрольної роботи,
- написання реферату по програмі курсу.
- контроль успішності на лабораторних заняттях, який включає в себе бали за виконання кожної лабораторної роботи.

3. **Лабораторні заняття.** Можливі наступні форми контролю знань:

- контроль успішності на лабораторних заняттях, який включає в себе бали за виконання кожної лабораторної роботи.

Додаток

Шкала оцінювання: Університету , національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка за ECTS	Оцінка за національною шкалою		
		іспит		залік
90-100	A	5	відмінно	зараховано
81-89	B	4	добре	
71-80	C			
61-70	D	3	задовільно	
51-60	E			
21–50	FX	2	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-20	F	2	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни