



Міністерство освіти і науки України  
Національний університет “Львівська політехніка”  
Кафедра автоматизованих систем управління

---

**Методичні вказівки**

до лабораторної роботи № 2

**«Моделювання бізнес-процесів у середовищі AllFusion® Process Modeler »**

з дисципліни

**«Проектування інформаційних систем»**

для студентів

спеціальності 122 “Комп’ютерні науки та інформаційні технології”

Львів-2018

Методичні вказівки до лабораторних робіт № 2 «Моделювання бізнес-процесів у середовищі AllFusion® Process Modeler» з дисципліни **“Проектування інформаційних систем”** для студентів спеціальності 122 “Комп’ютерні науки та інформаційні технології”  
Укл. Дорошенко А.В., Львів: Національний університет “Львівська політехніка”, 2018.

Методичні вказівки обговорено та схвалено на засіданні кафедри АСУ  
Протокол № \_\_\_\_\_ від «\_\_» \_\_\_\_\_ 2018 р.

Завідувач кафедрою АСУ \_\_\_\_\_ *Цмоць І.Г.*

Методичні вказівки обговорено та схвалено на засіданні методичної комісії  
базового напрямку підготовки

Протокол № \_\_\_\_\_ від «\_\_» \_\_\_\_\_ 2018 р.

## Лабораторна робота № 2

### **«Моделювання бізнес-процесів засобами AllFusion Process Modeler (Bpwin)»**

**Тема роботи:** Ознайомлення із середовищем AllFusion Process Modeler (Bpwin) та побудова IDEF0-діаграми.

**Мета роботи:** Освоїти моделювання бізнес-процесів у середовищі AllFusion Process Modeler, розробити моделі заданого бізнес-процесу у нотаціях IDEF0, DFD та IDEF3.

### **1. Теоретичні відомості**

*Технологія* створення інформаційних систем (ІС) ставить особливі вимоги до методик реалізації, програмних та інструментальних засобів, а саме:

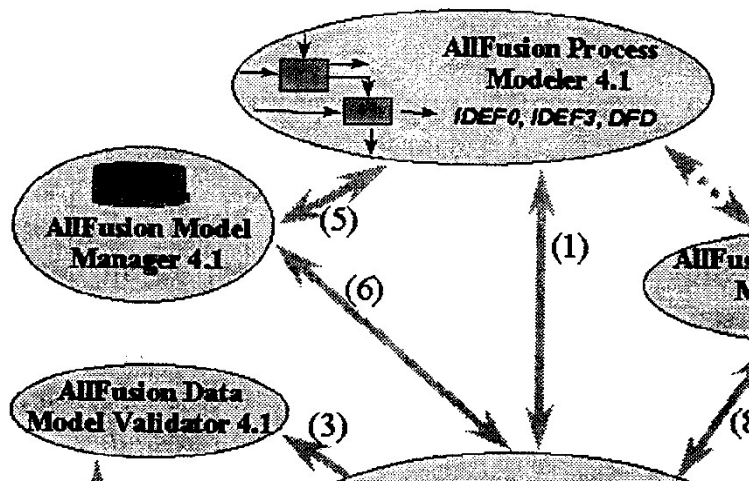
1. Реалізацію проектів по створенню ІС прийнято розбивати на стадії аналізу (перш ніж *створювати* ІС, необхідно зрозуміти й описати бізнес-логіку предметної області), проектування (необхідно визначити модулі та архітектуру майбутньої системи), безпосереднього кодування, тестування й супроводу. Відомо, що виправлення помилок, допущених на попередній стадії, обходиться приблизно в 10 раз дорожче, ніж на поточній, звідки випливає, що найбільш критичними є перші стадії проекту. *Тому надзвичайно важливо мати ефективні засоби автоматизації ранніх етапів реалізації проекту.*
2. Проект із створення складної ІС неможливо реалізувати одноосібно. Колективна робота суттєво відрізняється від індивідуальної, тому під час *реалізації великих проектів необхідно мати засоби координації й управління колективом розробників.*
3. Життєвий цикл створення складної ІС є співставним із очікуваним терміном: її експлуатації. Інакше кажучи, у сучасних умовах компанії перебудовують свої бізнес-процеси приблизно раз у два роки, стільки ж часу потрібно (якщо працювати в традиційній технології) для створення ІС. Може виявитися, що до моменту здачі ІС вона вже нікому не потрібна, оскільки компанія, що її замовила, змушена перейти на нову технологію роботи. *Отже, для створення ІС життєво необхідний інструмент, який значно (у кілька раз) зменшує час розробки ІС.*
4. Внаслідок досить тривалого життєвого циклу може виявитися, що в процесі створення системи зовнішні умови змінилися. Звичайне внесення змін у проект на пізніх етапах створення ІС - досить трудомісткий і дорогий процес. Тому для успішної реалізації великого проекту необхідно, щоб *інструментальні засоби, на яких він реалізується, були досить гнучкими до вимог, що змінюються.*

На сучасному ринку засобів розробки ІС досить багато систем, тією чи іншою мірою задовольняючих перерахованим вимогам. Зокрема, серед таких case-засобів варто відзначити: *AllFusion Modeling Suite 4.1* (Computer Associates), *Silverrun* (Silverrun technology), *Oracle Designer* (Oracle), *Rational Rose* (Rational Software) та ін.

Оскільки найбільш розповсюдженим та оптимальним за співвідношенням ціна/якість серед багатьох case-засобів є *AllFusion Modeling Suite 4.1* (відомий раніше під назвами *Bpwin 4.1* та *ERwin*) надалі для моделювання будемо використовувати саме його.

Функціональні можливості інструментальних засобів структурного моделювання ділових процесів будуть розглянуті на прикладі case-засобу AllFusion Process Modeler (Bpwin). Bpwin підтримує три методології моделювання: *функціональне моделювання (IDEF0)*; *опис бізнес-процесів (IDEF3)*; *діаграми потоків даних (DFD)*.

ли день по критерию стоимость/эффективность.



Загальна схема взаємодії інструментальних засобів AllFusion Modeling Suite 4.1

### Інструментальне середовище AllFusion Process Modeler (Bpwin)

Bpwin має досить простий і інтуїтивно зрозумілий *інтерфейс* користувача. При запуску AllFusion Process Modeler за замовчуванням з'являється основна *панель інструментів*, палітра інструментів (вид якої залежить від обраної нотації) і, у лівій частині, навігатор моделі - *Modelexplorer* (рис. 1).

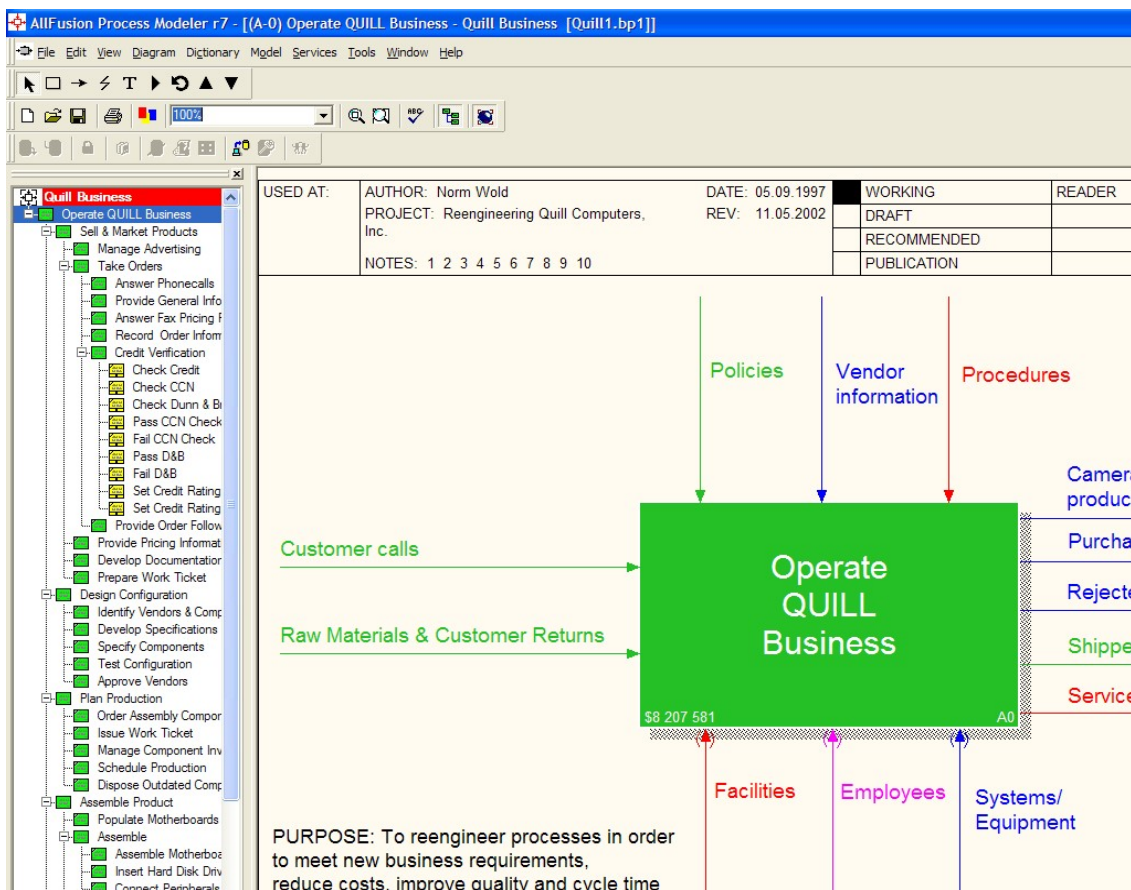
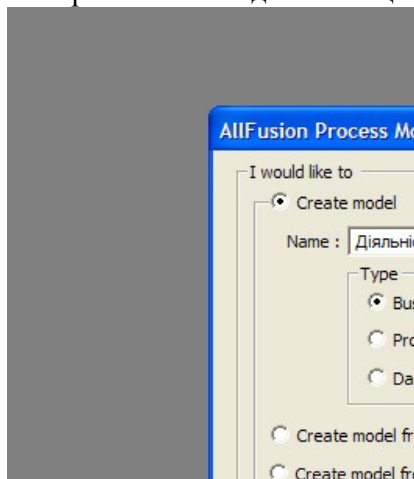


Рис.1. Інтегроване середовище розробки моделі AllFusion Process Modeler

Під час створення нової моделі виникає діалог, у якому слід вказати, чи буде модель створена заново або вона буде відкрита з файлу або з репозиторію Modelmart, потім ввести ім'я моделі й вибрати методологію, у якій буде побудована модель (рис. 2).

Як було зазначено вище, AllFusion Process Modeler підтримує три методології - *IDEF0*, *IDEF3* і *DFD*, кожна з яких вирішує свої специфічні завдання. В AllFusion Process Modeler можлива побудова змішаних моделей, тобто модель може містити одночасно діаграми як *IDEF0*, так і *IDEF3* и *DFD*. Вміст палітри інструментів змінюється автоматично, коли відбувається перемикання з однієї нотації на іншу.



**Рис.2.** Діалог створення моделі

Модель в AllFusion Process Modeler розглядається як сукупність *робіт*, кожна з яких оперує з деяким набором даних. *Робота* зображається у вигляді прямокутників, дані - у вигляді стрілок. Якщо клацнути по будь-якому об'єкту моделі лівою кнопкою миші, з'являється контекстне *меню*, кожний *пункт* якого відповідає редактору деякої властивості об'єкта.

### **Побудова моделі IDEF0**

На початкових *етапах створення ІС* необхідно зрозуміти, як працює організація, яку збираються автоматизувати. Керівник добре знає *роботу* в цілому, але не в змозі вникнути в деталі *роботи* кожного рядового співробітника. Рядовий співробітник добре знає, що діється на його робочому місці, але може не знати, як працюють колеги. Тому для опису *роботи* підприємства необхідно побудувати модель, яка буде адекватно відображати *предметну область* й містити в собі знання всіх учасників бізнес-процесів організації.

Найбільш зручною мовою моделювання бізнес-процесів є *IDEF0*, де система представляється як сукупність взаємодіючих *робіт* або функцій. Така чисто функціональна орієнтація є принциповою - функції системи аналізуються незалежно від об'єктів, якими вони оперують. Це дозволяє більш чітко змодельовати логіку й взаємодію процесів організації.

Процес моделювання системи в *IDEF0* починається зі створення *контекстної діаграми* – діаграми найбільш абстрактного рівня опису системи в цілому, що містить *визначення* суб'єкта моделювання, мети й точки зору на модель.

Під суб'єктом розуміється сама система, при цьому необхідно точно встановити, що входить у систему, а що лежить за її межами, інакше кажучи, визначити, що буде надалі розглядатися як компоненти системи, а що як зовнішній вплив. На *визначення суб'єкта системи* будуть суттєво впливати позиція, з якої розглядається система, і *мета моделювання* - питання, на які побудована модель повинна дати

відповідь. Інакше кажучи, на початку необхідно визначити область моделювання. Опис області як системи в цілому, так і її компонентів є основою побудови моделі. Хоча передбачається, що в ході моделювання область може коректуватися, вона повинна бути в основному сформульована від самого початку, оскільки саме область визначає напрямок моделювання. При формулюванні області необхідно враховувати два компоненти - широту й глибину. Широта має на увазі *визначення* границь моделі - що буде розглядатися усередині системи, а що зовні. Глибина визначає, на якому рівні деталізації модель є завершеною. При визначенні глибини системи необхідно пам'ятати про обмеження часу - трудомісткість побудови моделі росте в геометричній прогресії зі збільшенням глибини декомпозиції. Після визначення границь моделі передбачається, що нові об'єкти не повинні вноситися в систему, що моделюється.

## Мета моделювання

*Мета моделювання* визначається з відповідей на наступні питання:

- Чому цей процес повинен бути змодельований?
- Що повинна показувати модель?
- Що може одержати клієнт?

*Точка зору (Viewpoint).*

Під точкою зору розуміється перспектива, з якої спостерігалася система при побудові моделі. Хоча при побудові моделі враховуються думки різних людей, усі вони повинні дотримуватися єдиної точки зору на модель. Точка зору повинна відповідати меті й границям моделювання. Як правило, вибирається точка зору людини, відповідальної за *роботу*, що моделюється, в цілому.

*IDEF0-модель* припускає наявність чіткої сформульованої мети, єдиного суб'єкта моделювання й однієї точки зору. Для внесення області, мети й точки зору в модель *IDEF0* в AllFusion Process Modeler слід вибрати пункт меню Model/Model Properties діалог, що викликає, Model (рис. 3.). У закладці Purpose слід внести мету й точку зору, а в закладку Definition – визначення моделі й опис області.

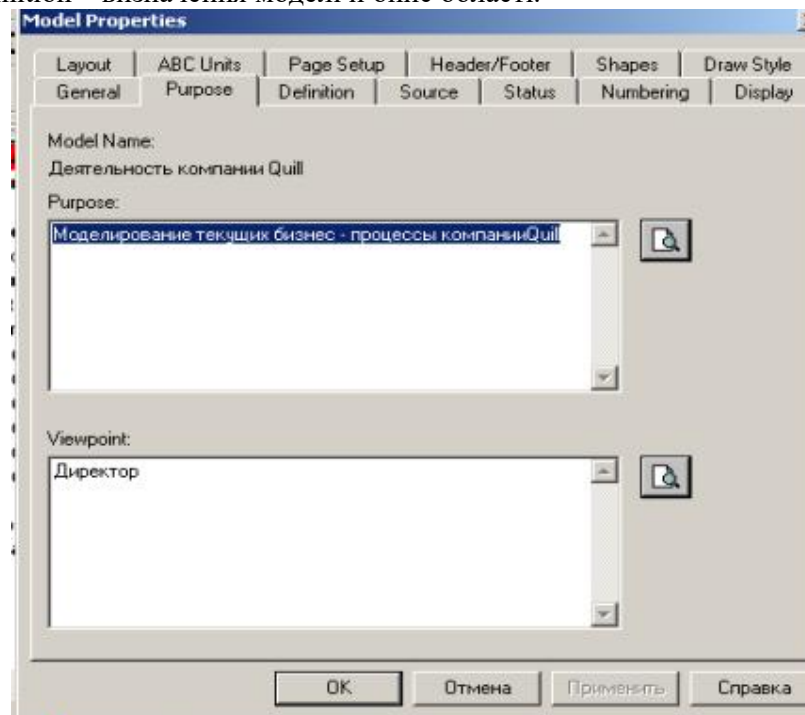


Рис. 3. Діалог визначення властивостей моделі

У закладці Status того ж діалогу можна описати статус моделі (чорновий варіант, робітник, остаточний і т.д.), час створення й останнього редагування (відслідковується надалі автоматично по системній даті). У закладці Source описуються джерела інформації для побудови моделі (наприклад, "Опитування експертів предметної області й аналіз документації"). Закладка General служить для внесення імені проекту й моделі, імені й ініціалів автора й тимчасових рамок моделі - AS-IS і TO-BE.

### Моделі AS-IS і TO-BE.

Зазвичай спочатку будується модель існуючої організації *роботи* - AS-IS (як є). Аналіз *функціональної моделі* дозволяє зрозуміти, де є найбільш слабкі місця, у чому будуть полягати переваги нових бізнес-процесів і наскільки глибоким змінам піддається існуюча *структура організації бізнесу*. Деталізація бізнес-процесів дозволяє виявити недоліки організації навіть там, де функціональність на перший погляд видається очевидною. Знайдені в моделі AS-IS недоліки можна виправити при створенні моделі TO-BE (як буде) – моделі нової організації бізнес-процесів.

Технологія *проектування ІС* передбачає спочатку створення моделі AS-IS, її аналіз і поліпшення бізнес-процесів, тобто створення моделі TO-BE, і тільки на основі моделі TO-BE будується модель даних, прототип і потім остаточний варіант ІС.

Іноді поточна AS-IS і майбутня TO-BE моделі відрізняються дуже сильно, внаслідок чого перехід від початкового до кінцевого стану стає неочевидним. У цьому випадку необхідна третя модель, що описує процес переходу від початкового до кінцевого стану системи, оскільки такий перехід – це теж бізнес-процес.

Результат опису моделі можна одержати у звіті Model Report. Діалог настроювання звіту по моделі викликається з пункту меню Tools/Reports/Model Report.

У діалозі настроювання слід вибрати необхідні поля, при цьому автоматично відображається черговість виводу інформації у звіт (рис.4).

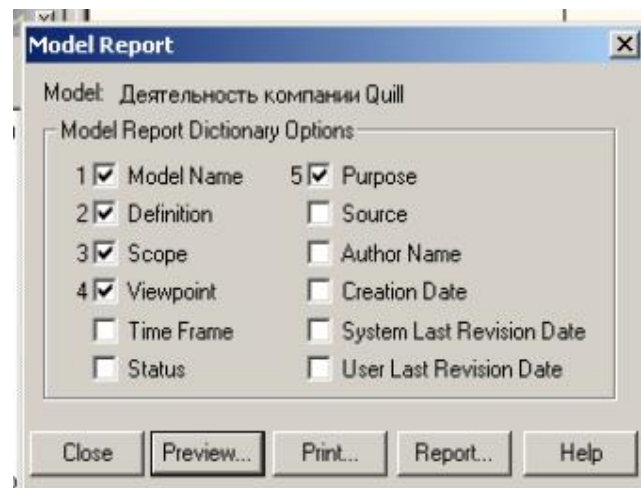
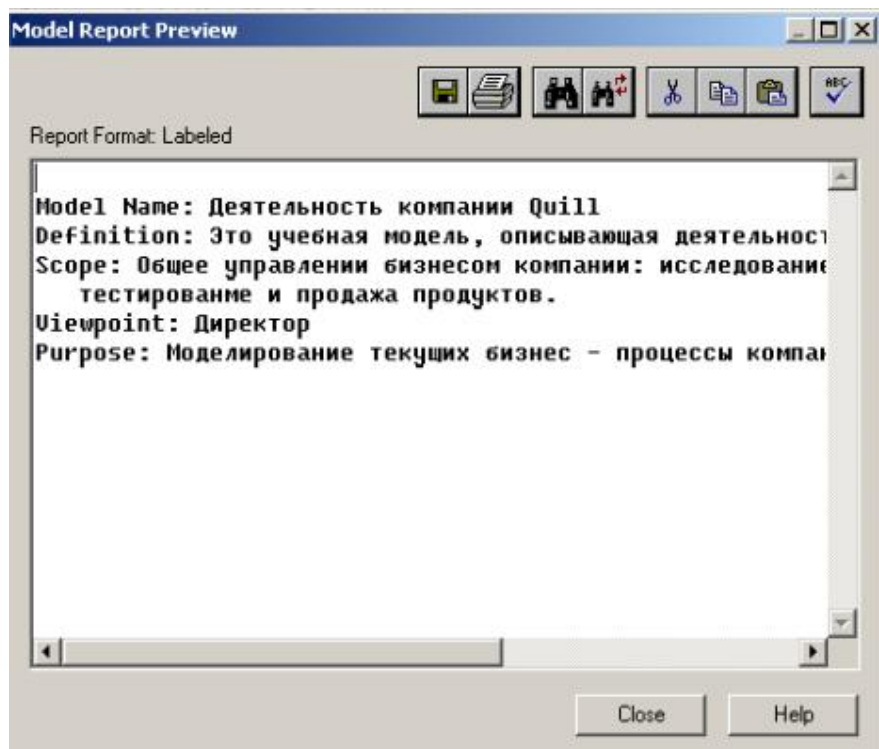


Рис. 4. Діалогове вікно для формування звіту по моделі

На рис.5. наведено звіт, сформований за вищевказаними полями.





**Рис. 5.** Попередній перегляд звіту

Основу методології *IDEF0* становить графічна мова опису бізнес-процесів. Модель у нотації *IDEF0* є сукупністю ієрархічно впорядкованих і взаємозалежних діаграм. Кожна діаграма є одиницею опису системи й розташовується на окремому аркуші.

Модель *може* містити чотири типи діаграм:

- *контекстну діаграму* (у кожній моделі може бути тільки одна *контекстна діаграма*);
- *діаграми декомпозиції*;
- *діаграми дерева вузлів*;
- *діаграми тільки для експозиції (FEO)*.

*Контекстна діаграма* є вершиною деревоподібної структури діаграм і являє собою найбільш загальний опис системи і її взаємодії із зовнішнім середовищем. Після опису системи в цілому проводиться розбивка її на великі фрагменти. Цей процес називається *функціональною декомпозицією*, а діаграми, які описують кожен фрагмент і взаємодію фрагментів, називаються *діаграмами декомпозиції*. Після декомпозиції *контекстної діаграми* проводиться декомпозиція кожного великого фрагмента системи на більш дрібні і так далі, до досягнення потрібного рівня подробности опису. Після кожного сеансу декомпозиції проводяться сеанси експертизи – експерти предметної області вказують на відповідність реальних бізнес-процесів створеним діаграмам. Знайдені невідповідності виправляються, і тільки після проходження експертизи без зауважень можна приступати до наступного сеансу декомпозиції. Так досягається відповідність моделі реальним бізнес-процесам на кожному рівні моделі. Синтаксис опису системи в цілому й кожного її фрагмента однаковий у всій моделі.

*Діаграма дерева вузлів* показує ієрархічну залежність *робіт*, але не взаємозв'язки між *роботами*. *Діаграм дерев вузлів* може бути в моделі як завгодно багато, оскільки дерево може бути побудоване на довільну глибину й не обов'язково з кореня.

*Діаграми для експозиції (FEO)* будуються для ілюстрації окремих фрагментів моделі, для ілюстрації альтернативної точки зору, або для спеціальних цілей.

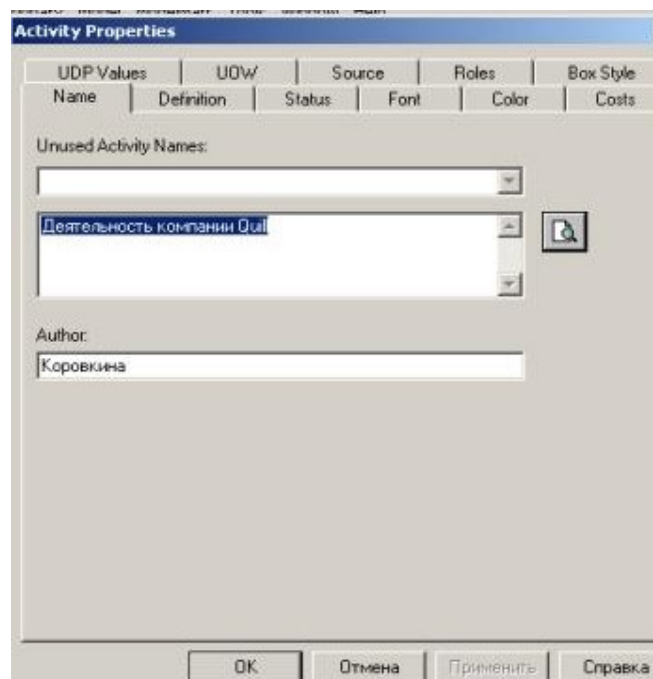


**Роботи (Activity)** позначають поійменовані процеси, функції або завдання, які відбуваються протягом певного часу й мають розпізнавані результати. *Роботи* зображаються у вигляді прямокутників. Усі *роботи* повинні бути названі й визначені. Ім'я *роботи* повинне бути виражене іменником, що позначає дію (наприклад, "Діяльність компанії", "Приймання замовлення" і т.д.). *Робота* "Діяльність компанії" може мати, наприклад визначення, що впливає: "Це навчальна модель, що описує діяльність компанії". При створенні нової моделі (меню File/New) автоматично створюється *контекстна діаграма* з єдиною *роботою*, що зображує систему в цілому (рис.6.).



**Рис. 6.** Приклад контекстної діаграми

Для внесення імені *роботи* слід клацнути по *роботі* правою кнопкою миші, вибрати в меню Name Editor і в діалозі, що з'явився, ввести ім'я роботи. Для опису інших властивостей *роботи* служить діалог Activity Properties (рис. 7).



**Рис. 7.** Редактор завдання властивостей роботи

Діаграми *декомпозиції* містять споріднені *роботи*, тобто дочірні *роботи*, що мають спільну батьківську *роботу*. Для створення *діаграми декомпозиції* слід клацнути по кнопці ■ на панелі інструментів.

Виникає діалог Activity Box Count (рис. 8), в якому слід вказати нотацію нової діаграми й кількість *робіт* на ній. Зупинимося поки на нотації *IDEF0* і клацнемо на ОК. З'являється діаграма декомпозиції (рис. 9). Припустимий інтервал числа *робіт* - 2-8. Робити декомпозицію *роботи* на одну *роботу* не має змісту, діаграми з кількістю *робіт* більше восьми виходять перенасиченими й погано читаються. Для забезпечення наочності й кращого розуміння процесів, що моделюються, рекомендується використовувати від трьох до шести блоків на одній діаграмі.



Рис. 8. Діалог Activity Box Count

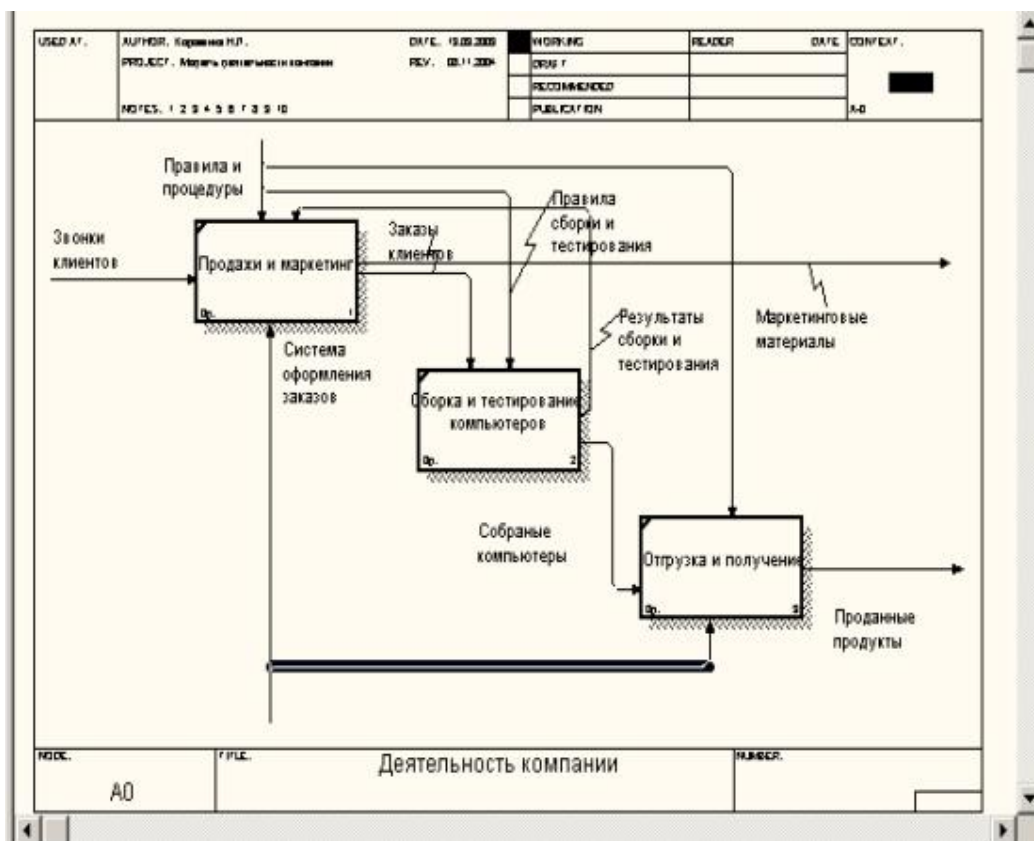


Рис. 9. Приклад діаграми декомпозиції

Якщо виявляється, що кількість *робіт* недостатня, то *роботу* можна додати в діаграму, клацнувши спочатку по кнопці на палітрі інструментів, а потім по вільному місці на діаграмі.

*Роботи* на діаграмах декомпозиції зазвичай розташовуються по діагоналі від лівого верхнього кута до правого нижнього.

Такий порядок називається порядком домінування. Згідно із цим принципом розташування в лівому верхньому куті міститься найважливіша *робота* або *робота*, виконувана за часом першою. Далі вправо вниз розташовуються менш важливі або виконувані пізніше *роботи*. Таке розміщення полегшує читання діаграм, крім того, на ньому ґрунтується поняття взаємозв'язків *робіт* (див. нижче).

Кожна з *робіт* на *діаграмі декомпозиції* може бути у свою чергу декомпонована. На *диаграмме декомпозиции работы* нумеруються автоматично зліва направо. Номер *роботи* показується у правому нижньому куті. В лівому верхньому куті зображається невелика діагональна риска, яка показує, що ця *робота* не була декомпонована. Так, на рис. 9 все *роботи* ще не були декомпоновані.

*Стрілки* (Arrow) описують взаємодії *робіт* і являють собою якусь інформацію, виражену іменниками. (Наприклад, "Дзвінки клієнтів", "Правила й процедури", "Бухгалтерська система".)

В *IDEF0* розрізняють п'ять типів стрілок:

**1. Вхід ( Input )** - матеріал або інформація, які використовуються або перетворюються *роботою* для одержання результату (виходу). Допускається, що *робота* може не мати жодної *стрілки* входу.

Кожен тип стрілок повинен підходити до визначеної сторони прямокутника, що зображує *роботу*, або виходити з неї.

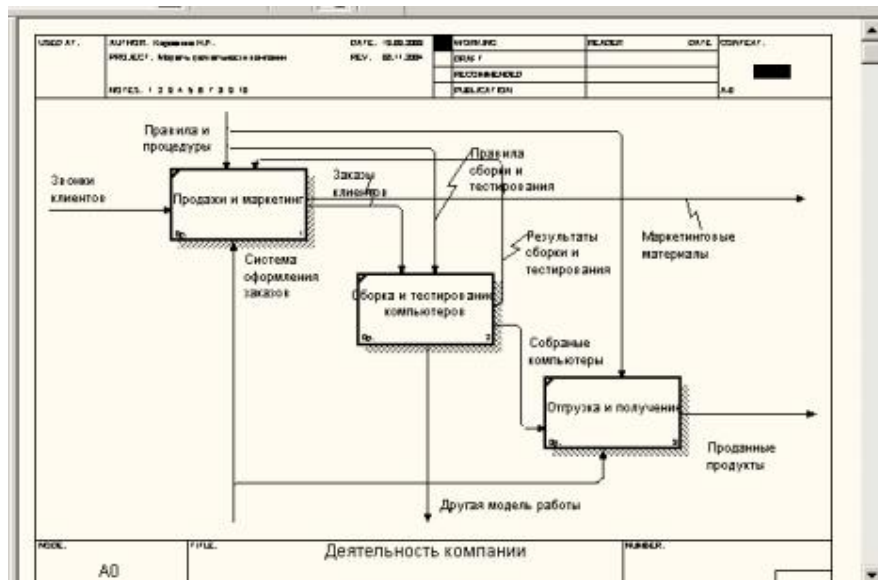
*Стрілка* входу рисується як вхідна в ліву грань *роботи*. При описі технологічних процесів (для цього й був придуманий *IDEF0*) не виникає проблем визначення входів. Дійсно, "Дзвінки клієнтів" на рис. 6 - це щось, що переробляється в процесі "Діяльність компанії" для одержання результату. При моделюванні ІС, коли *стрілками* є не фізичні об'єкти, а дані, не все так очевидно. Наприклад, при "Прийманні пацієнта" карта пацієнта може бути як на вході, так і на виході, тим часом якість цих даних міняється. Інакше кажучи, у нашому прикладі для того, щоб виправдати своє призначення, *стрілки* входу й виходу повинні бути точно визначені для того, щоб указати на те, що дані дійсно були перероблені (наприклад, на виході - "Заповнена карта пацієнта"). Дуже часто складно визначити, чи є дані входом або керуванням. У цьому випадку підказкою може служити інформація про те, переробляються/ чи змінюються дані в *роботі* чи ні. Якщо змінюються, то, швидше за все, це вхід, якщо ні - керування.

**2. Керування ( Control )** – правила, стратегії, процедури або стандарти, якими керується *робота*. Кожна *робота* повинна мати хоча б одну *стрілку* керування. *Стрілка* керування зображається як вхідна у верхню грань *роботи*. На рис. 6 *стрілка* "Правила й процедури" є керуючою для *роботи* "Діяльність компанії". Керування впливає на *роботу*, але не перетворюється *роботою*. Якщо мета *роботи* – змінити процедуру або стратегію, то така процедура або стратегія буде для *роботи* входом. У випадку виникнення невизначеності в статусі *стрілки* (керування або вхід) рекомендується малювати *стрілку* керування.

**3. Вихід ( Output )** – матеріал або інформація, які проводяться *роботою*. Кожна *робота* повинна мати хоча б одну *стрілку* виходу. *Робота* без результату не має змісту й не повинна моделюватися. *Стрілка* виходу зображається як вихідна із правої грані *роботи*. На рис. 6 *стрілки* "Маркетингові матеріали" і "Продані продукти" є виходом для *роботи* "Діяльність компанії".

**4. Механізм ( Mechanism )** – ресурси, які виконують *роботу*, наприклад персонал підприємства, верстати, пристрої і т.д. *Стрілка* механізму зображається як вхідна в нижню грань *роботи*. На рис. 6 *стрілка* "Бухгалтерська система" є механізмом для *роботи* "Діяльність компанії". Аналітик має право вирішити, що *стрілки* механізму можуть не зображатись в моделі.



**5. Виклик ( Call )** – спеціальна *стрілка*, що вказує на іншу модель *роботи*. *Стрілка* виклику зображається як вихідна з нижньої грані *роботи*. На рис. 10 *стрілка* "Інша модель *роботи*" є викликом для *роботи* "Виготовлення виробу". *Стрілка* виклику використовується для того, щоб вказати, що деяка *робота* виконується за межами системи, що моделюється. В AllFusion Process Modeler *стрілки* виклику використовуються в механізмі злиття й поділу моделей.



**Рис. 10.** *Стрілка* виклику, що з'являється при розщепленні моделі

**Граничні *стрілки*.** *Стрілки* на контекстній діаграмі служать для опису взаємодії системи з навколишнім світом. Вони можуть починатися на межі діаграми й закінчуватися в *роботі*, або навпаки. Такі *стрілки* називаються граничними.

Для внесення граничної *стрілки* входу необхідно:

- клацнути по кнопці із символом *стрілки* ;
- у палітрі інструментів перевести курсор до лівої сторони екрана, поки не з'явиться початкова штрихова смужка;
- клацнути один раз по смужці (звідки виходить *стрілка* ) і ще раз у лівій частині *роботи* з боку входу (де закінчується *стрілка* );
- повернутися в палітру інструментів і вибрати опцію редагування *стрілки* .
- клацнути правою кнопкою миші на лінії *стрілки*, у випливаючому меню вибрати Name і додати ім'я *стрілки* в закладці Name діалогу IDEF0 Arrow Properties.

*Стрілки* керування, входу, механізму й виходу зображаються аналогічно. Імена створених стрілок (рис. 11) автоматично заносяться в словник Arrow Dictionary.

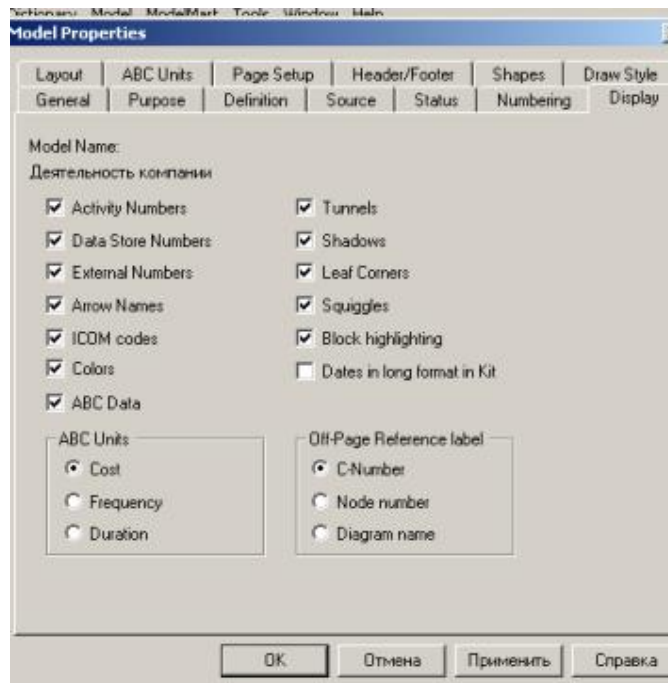


Рис. 11. Діалог IDEF0 Arrow Properties

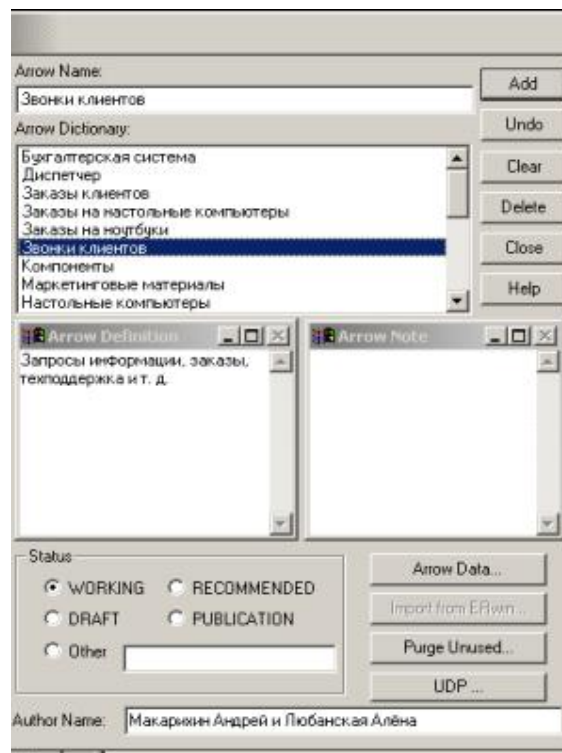
**ICOM-Коди.** Діаграма декомпозиції призначена для деталізації *роботи*. На відміну від моделей, що відображають *структуру організації*, *робота* на діаграмі верхнього рівня в IDEF0 – це не елемент керування *роботами нижчого рівня*. *Роботи* нижнього рівня – це те саме, що *роботи* верхнього рівня, але в більш детальному викладі. Як наслідок цього границі *роботи* верхнього рівня – це те саме, що границі *діаграми декомпозиції*. ICOM (аббревіатура від Input, Control, Output і Mechanism) - коди, призначені для ідентифікації граничних стрілок. Код ICOM містить префікс, відповідний до типу *стрілки* (I, 3, Про або M), і порядковий номер.

AllFusion Process Modeler вносить Icom-Коди автоматично. Для відображення Icom-Кодів слід включити опцію ICOM codes на закладці Display діалогу Model Properties (меню Model/Model Properties) (рис. 12).

Словник **стрілок** редагується за допомогою спеціального редактора Arrow Dictionary Editor, у якому визначається *стрілка* й вноситься коментар, що її описує (рис.13). Словник стрілок вирішує дуже важливе завдання. Діаграми створюються аналітиком для того, щоб провести сеанс експертизи, тобто обговорити діаграму з *фахівцем предметної області*. У будь-якій предметній області формується професійний жаргон, при чому дуже часто жаргонні вирази мають нечіткий сенс і сприймаються різними фахівцями по-різному. У той самий час аналітик – автор діаграм повинен уживати ті вирази, які найбільш зрозумілі експертам. Оскільки формальні визначення часто складні для сприйняття, аналітик змушений вживати професійний жаргон, а щоб не виникло неоднозначних трактувань, у словнику стрілок кожному поняттю можна дати розширене й, якщо це необхідно, формальне визначення.



*Рис. 12. Включення опції ICOM codes на закладці Display*

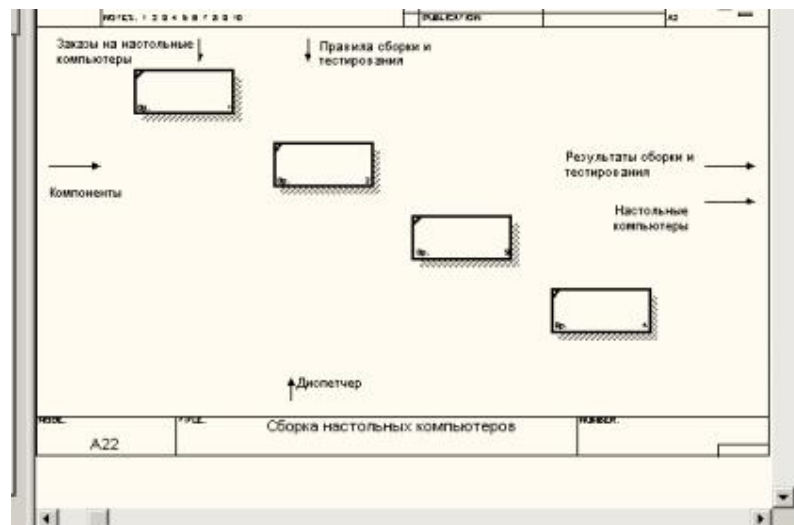


*Рис. 13. Редагування словника стрілок*

Вміст словника стрілок можна роздрукувати у вигляді звіту (меню Tools/ Report /Arrow Report...) і одержати тлумачний словник термінів предметної області, що використовуються в моделі.

**Незв'язані граничні стрілки (unconnected border arrow).** Під час декомпозиції роботи її вхідні та вихідні стрілки (крім стрілки виклику) автоматично з'являються на діаграмі декомпозиції (міграція стрілок), але при цьому не стосуються роботи. Такі стрілки називаються незв'язаними й сприймаються в AllFusion Process Modeler як синтаксична помилка.

На рис. 14 наведено фрагмент діаграми декомпозиції з незв'язаними стрілками, який згенерував AllFusion Process Modeler під час декомпозиції роботи **"Складання настільних комп'ютерів"** (див. рис. 9). Для зв'язування стрілок входу, керування або механізму необхідно перейти в режим редагування стрілок, клацнути по наконечникові стрілки й потім по відповідному сегменту роботи. Для зв'язування стрілки виходу необхідно перейти в режим редагування стрілок, клацнути по сегменту виходу роботи й потім по стрілці.



**Рис. 14.** Приклад незв'язаних стрілок

**Внутрішні стрілки.** Для зв'язку робіт між собою використовуються внутрішні стрілки, тобто стрілки, які не торкаються границі діаграми, починаються в одній і закінчуються в іншій роботі.

Для малювання внутрішньої стрілки необхідно в режимі малювання стрілок клацнути по сегменту (наприклад, виходу) однієї роботи й потім по сегменту (наприклад, входу) іншої. В IDEF0 розрізняють п'ять типів зв'язків робіт:

**1. Зв'язок по входу (output-input),** коли стрілка виходу вищої роботи (далі - просто вихід) направляється на вхід нижчестоящої (наприклад, на рис. 15 стрілка **"Зібрані комп'ютери"** зв'язує роботи **"Складання й тестування комп'ютерів"** і **"Відвантаження й одержання"**).



**Рис. 15.** Зв'язок по входу



**2. Зв'язок по керуванню ( output-control ),** коли вихід вищої *роботи* направляється на керування нижчестоящої. Зв'язок по керуванню показує домінування вищої *роботи*. Дані або об'єкти виходу вищої *роботи* не змінюються в роботі нижнього рівня. На рис. 16 стрілка "Замовлення клієнтів" зв'язує *роботи* "Продаж і маркетинг" і "Складання й тестування комп'ютерів".

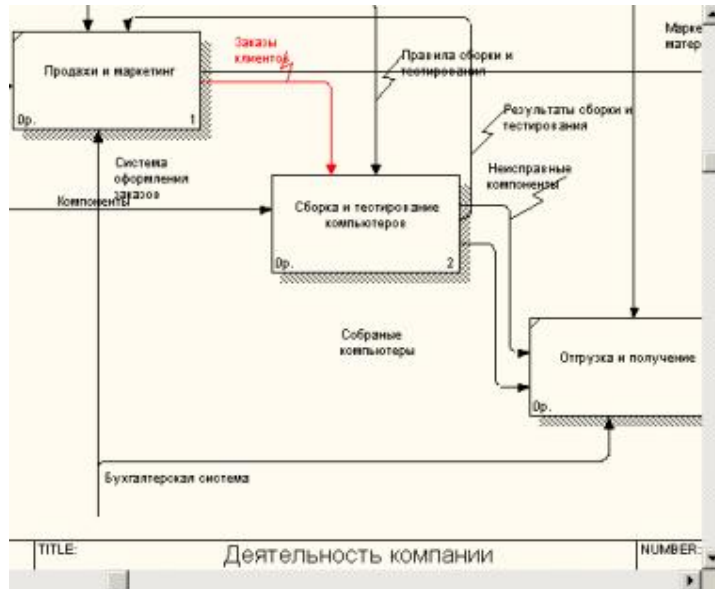


Рис. 16. Зв'язок по керуванню

**3. Зворотний зв'язок за входом ( output-input feedback ),** коли вихід *роботи* нижнього рівня направляється на вхід *роботи* вищого рівня. Такий зв'язок, як правило, використовується для опису циклів. На рис. 17 стрілка "Результаты тестирования" зв'язує *роботи* "Тестирование компьютеров" і "Видстежения розклада й керування складанням і тестуванням".

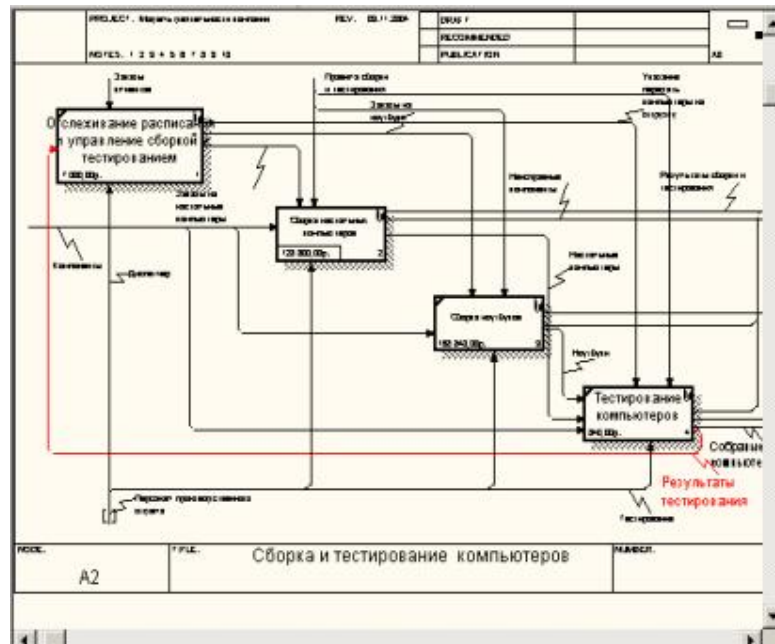
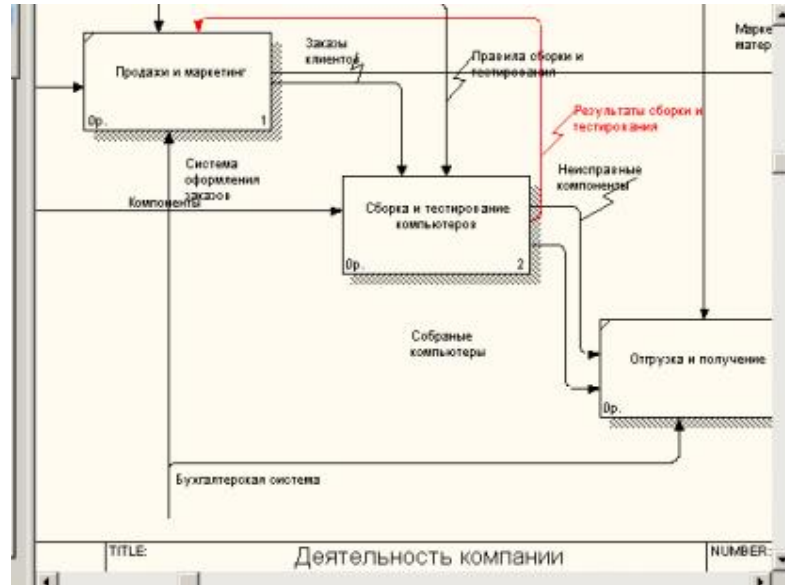


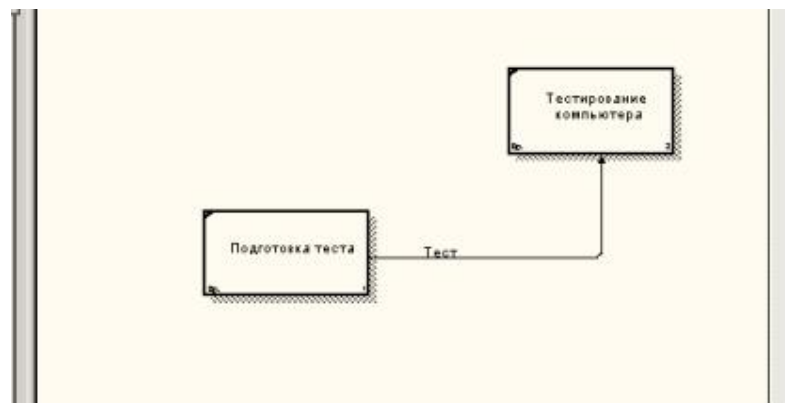
Рис. 17. Зворотний зв'язок по входу

**4. Зворотний зв'язок за керуванням ( output-control feedback )**, коли вихід *роботи* нижнього рівня направляється на керуючий вхід *роботи* вищого рівня (стрілка "Результати складання й тестування", рис. 18). Зворотний зв'язок за керуванням часто свідчить про ефективність бізнесу-процесу. На рис. 18 обсяг продажів може бути підвищений шляхом безпосереднього регулювання процесів складання й тестування комп'ютерів (виходу) *роботи* "Складання й тестування комп'ютерів".



**Рис. 18.** Зворотний зв'язок за керуванням

**5. Зв'язок вихід-механізм ( output-mechanism )**, коли вихід однієї *роботи* направляється на механізм іншої. Цей взаємозв'язок використовується рідше за інші і показує, що одна *робота* готує ресурси, необхідні для проведення іншої *роботи* (рис. 19).



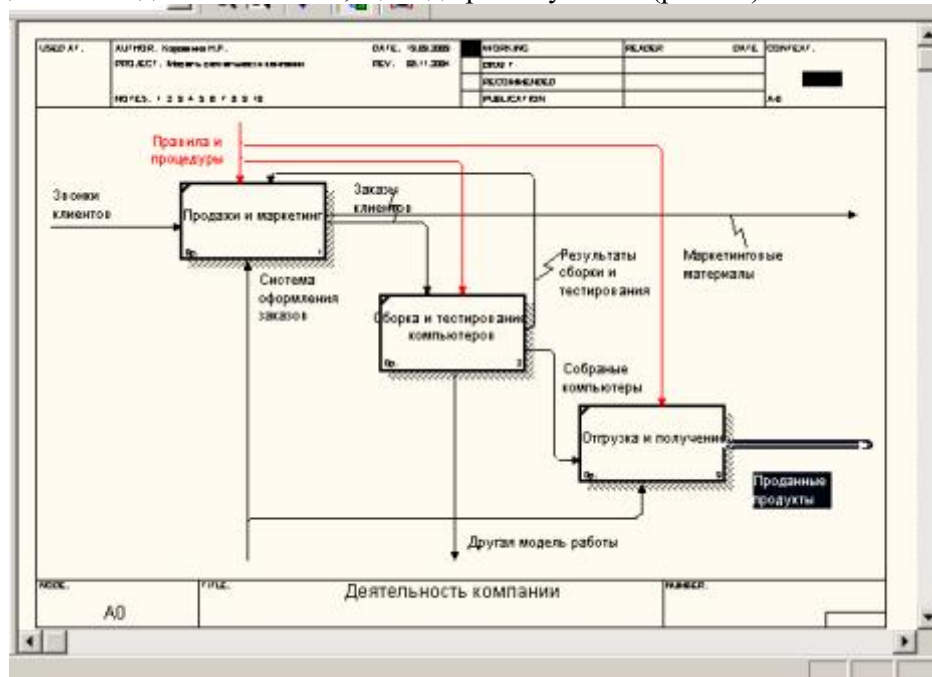
**Рис. 19.** Зв'язок вихід-механізм

**Явні стрілки.** Явна стрілка має джерелом одну-єдину *роботу* й призначенням теж одну-єдину *роботу*.

**Стрілки, що розгалужуються та стрілки, що зливаються.** Ті самі дані або об'єкти, породжені одною і тою самою *роботою*, можуть використовуватись відразу в декількох інших *роботах*. З іншого боку, стрілки, породжені в різних *роботах*, можуть являти собою однакові або однорідні дані або об'єкти, які надалі використовуються або

переробляються в одному місці. Для моделювання таких ситуацій в *IDEF0* використовуються *стрілки, що зливаються* та *стрілки, що розгалужуються*. Для *розгалуження стрілки* потрібно в режимі редагування *стрілки* клацнути по фрагменту *стрілки* й по відповідному сегменту *роботи*. Для злиття двох стрілок виходу потрібно в режимі редагування *стрілки* спочатку клацнути по сегменту виходу *роботи*, а потім по відповідному до фрагменту *стрілки*.

Зміст стрілок, що розгалужуються або зливаються, передається іменуванням кожної гілки стрілок. Існують певні правила іменування таких стрілок. Розглянемо їх на прикладі стрілок, що розгалужуються. Якщо *стрілка* іменована до розгалуження, а після розгалуження жодна з гілок не іменована, то мається на увазі, що кожна нова розгалужена гілка моделює ті ж дані або об'єкти, що й до розгалуження (рис. 20).



**Рис. 20.** Приклад іменування стрілки, що розгалужується

Якщо *стрілка* іменована до розгалуження, а після розгалуження яка-небудь із гілок теж іменована, то мається на увазі, що ці гілки відповідають іменуванню. Якщо при цьому яка-небудь гілка після розгалуження залишилася неіменованою, то мається на увазі, що вона моделює ті ж дані або об'єкти, що й гілка до розгалуження (рис. 21).

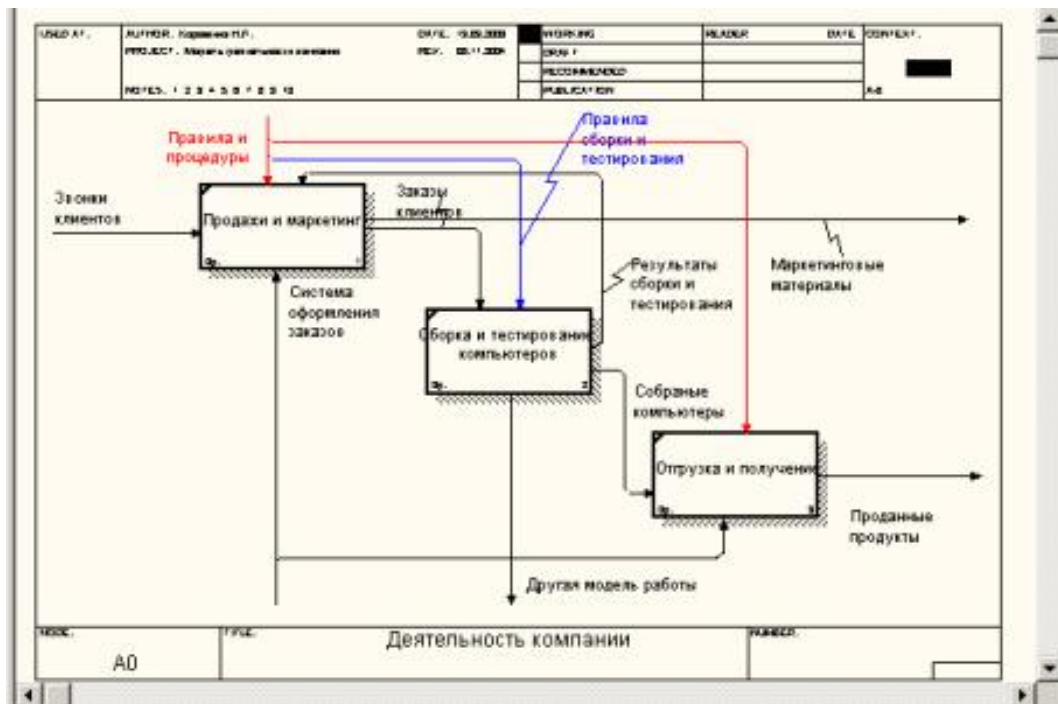


Рис. 21. Приклад іменування стрілки, що розгалужується

Неприпустима ситуація, коли *стрілка* до розгалуження не іменована, а після розгалуження не іменована яка-небудь із гілок. AllFusion Process Modeler визначає таку *стрілку* як синтаксичну помилку.

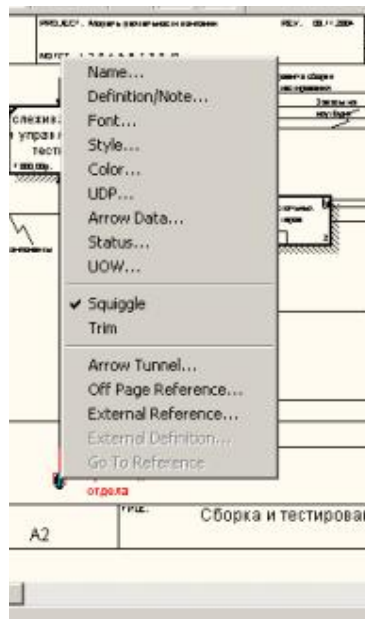
Правила іменування стрілок, що зливаються, повністю аналогічні – помилкою буде *вважатися* стрілка, яка після злиття не іменована, а до злиття не іменована яка-небудь із її гілок. Для іменування окремої гілки стрілок, що розгалужуються або зливаються, слід виділити на діаграмі тільки одну гілку, після чого викликати редактор імені й *присвоїти* ім'я стрілці. Це ім'я буде відповідати тільки виділеній гілкам.

**Тунелювання стрілок.** Нові внесені граничні *стрілки* на *діаграмі декомпозиції* нижнього рівня зображуються у квадратних дужках і автоматично не з'являються на діаграмі верхнього рівня (рис. 22).



Рис. 22. Недозволена (unresolved) стрілка

Для їх "перетаскування" наверх потрібно клацнути правою кнопкою миші по квадратних дужках граничної стрілки й у контекстному меню вибрати команду *Arrow Tunnel*(рис. 23).



**Рис. 23.** Вибір команди з контекстного меню

З'являється діалог Border Arrow Editor (рис. 24).

Якщо клацнути по кнопці *Resolve Border Arrow*, стрілка мігрує на діаграму верхнього рівня, якщо по кнопці *Change To Tunnel* - стрілка буде тунельована й не потрапить на іншу діаграму. Тунельна стрілка зображається із круглими дужками на кінці (рис. 25).



**Рис. 24.** Діалог Border Arrow Editor

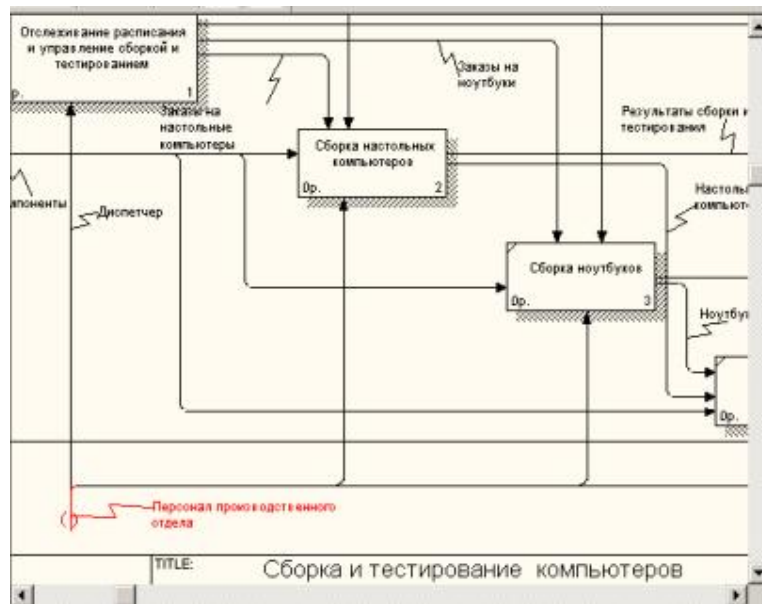


Рис. 25. Типи тунелювання стрілок

Тунелювання може бути застосоване для зображення малозначимих стрілок. Якщо на якій-небудь діаграмі нижнього рівня необхідно зобразити малозначимі дані або об'єкти, які не обробляються або не використовуються *роботами* на поточному рівні, то їх необхідно направити на вищий рівень (на батьківську діаграму). Якщо ці дані не використовуються на батьківській діаграмі, їх потрібно направити ще вище, і т.д. У результаті малозначима *стрілка* буде зображена на всіх рівнях і утруднить читання всіх діаграм, на яких вона присутня. Виходом є *тунелювання стрілки* на самому нижньому рівні. Таке *тунелювання* називається "не-в-батьківській-діаграмі".

Іншим прикладом *тунелювання* може бути ситуація, коли *стрілка* механізму мігрує з верхнього рівня на нижній, причому на нижньому рівні цей механізм використовується однаково у всіх *роботах* без винятку. (Передбачається, що не потрібно деталізувати *стрілку* механізму, тобто *стрілка* механізму на дочірній *роботі* іменована до розгалуження, а після розгалуження гілки не мають власного імені). У цьому випадку *стрілка* механізму на нижньому рівні може бути вилучена, після чого на батьківській діаграмі вона може бути тунельована, а в коментарі до *стрілки* або в словнику можна вказати, що механізм буде використовуватися у всіх *роботах* дочірньої *діаграми декомпозиції*. Таке *тунелювання* називається "не- в- дочірній-роботі" (рис. 25).

**Нумерація робіт і діаграм.** Усі *роботи* моделі нумеруються. Номер складається із префікса й числа. Може бути використаний префікс будь-якої довжини, але зазвичай використовують префікс А. Контекстна (коренева) *робота* дерева має номер А0. *Роботи* і декомпозиції А0 мають номери А1, А2, А3 тощо. *Роботи* декомпозиції нижнього рівня мають номер батьківської *роботи* й черговий порядковий номер, наприклад *роботи* декомпозиції А3 будуть мати номери А31, А32, А33, А34 і т.д. *Роботи* утворюють ієрархію, де кожна *робота* може мати одну батьківську й кілька дочірніх *робіт*, утворюючи дерево. Таке дерево називають деревом вузлів, а вищеописану нумерацію - нумерацією по вузлах. Діаграми IDEF0 мають подвійну нумерацію. По-перше, діаграми мають номери по вузлу. Контекстна *діаграма* завжди має номер А-0, декомпозиція *контекстної діаграми* - номер А0, інші *діаграми декомпозиції* - номери по відповідному до вузла (наприклад, А1, А2, А21, А213 і т.д.). AllFusion Process Modeler автоматично підтримує нумерацію по вузлах, тобто при проведенні декомпозиції



створюється нова діаграма і їй автоматично привласнюється відповідний номер. У результаті проведення експертизи діаграми можуть уточнюватися й змінюватися, отже, можуть бути створені різні версії однієї й тієї ж (з погляду її розташування в дереві вузлів) *діаграми декомпозиції*. AllFusion Process Modeler дозволяє мати в моделі тільки одну *діаграму декомпозиції* в даному вузлі. Колишні версії діаграми можна зберігати у вигляді паперової копії або як *FEO -Діаграму*. (На жаль, при створенні *FEO -діаграм* відсутня можливість відкату, тобто з діаграми можна одержати декомпозиції *FEO*, але не навпаки.) У кожному разі слід відрізняти різні версії однієї й тієї ж діаграми. Для цього існує спеціальний номер - C-number, який повинен привласнюватися автором моделі вручну. C-number – це довільний рядок, але рекомендується дотримуватися стандарту, коли номер складається з буквеного префікса й порядкового номера, причому в якості префікса використовуються ініціали автора діаграми, а порядковий номер відслідковується автором вручну, наприклад MCB00021.

### Діаграми дерева вузлів і FEO

*Діаграма дерев вузлів* показує ієрархію *робіт* у моделі й дозволяє розглянути всю модель цілком, але не показує взаємозв'язку між *роботами* (рис. 26). Процес створення моделі *робіт* є ітераційним, отже, *роботи* можуть змінювати своє розташування в дереві вузлів багаторазово. Щоб не заплутатися й перевірити спосіб декомпозиції, необхідно після кожної зміни створювати *діаграму дерева вузлів*. Втім, AllFusion Process Modeler має потужний інструмент навігації по моделі - Model Explorer, який дозволяє представити ієрархію *робіт* і діаграм у зручному й компактному виді.

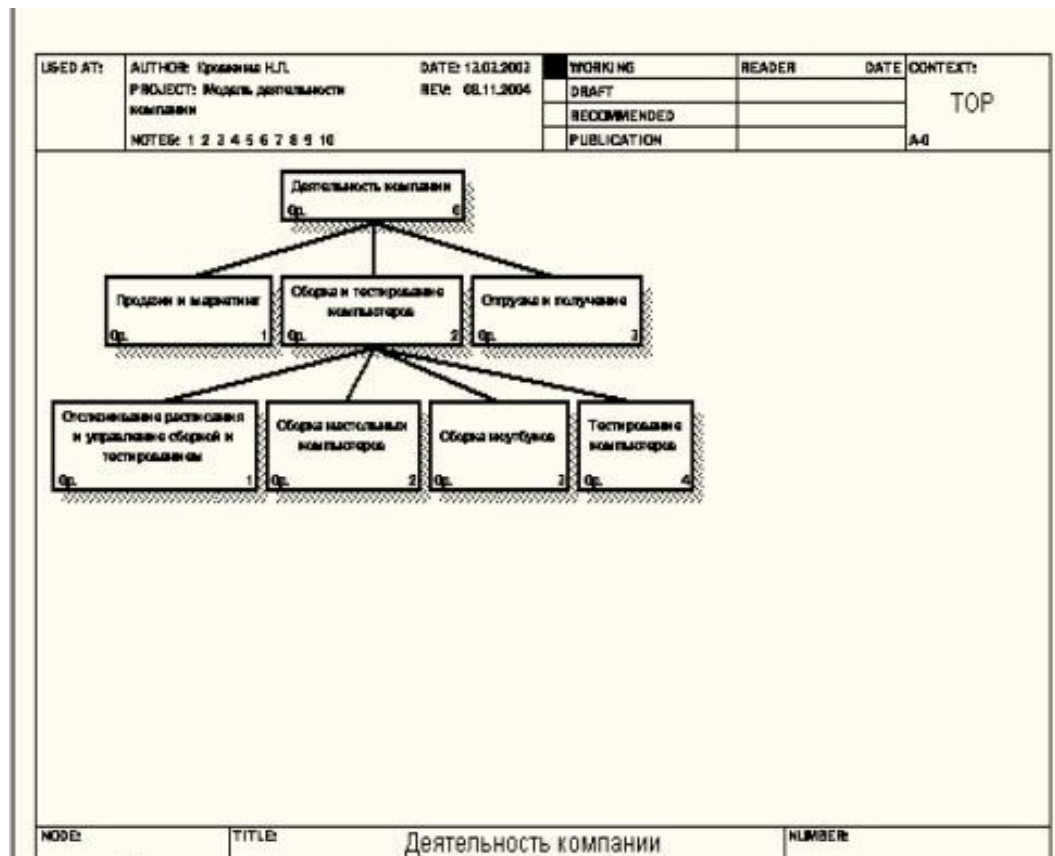
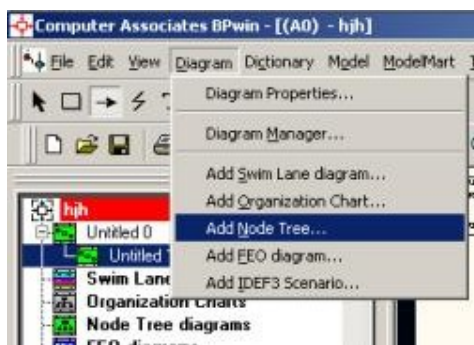


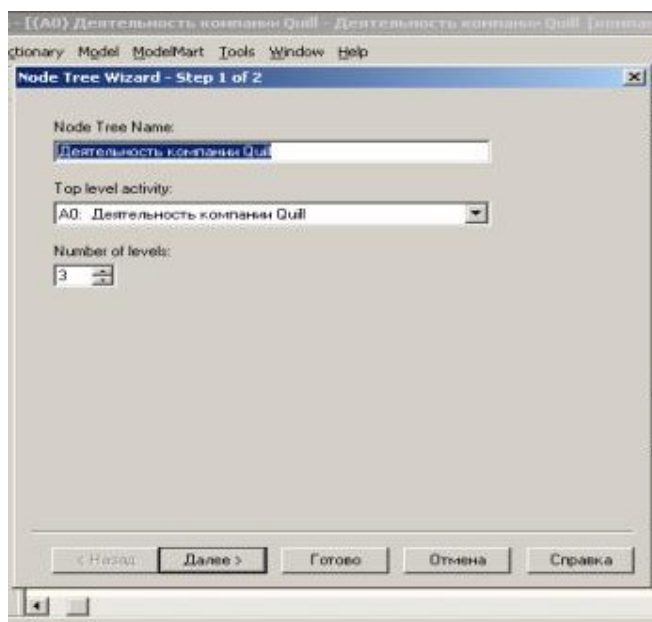
Рис. 26. Діаграма дерева вузлів





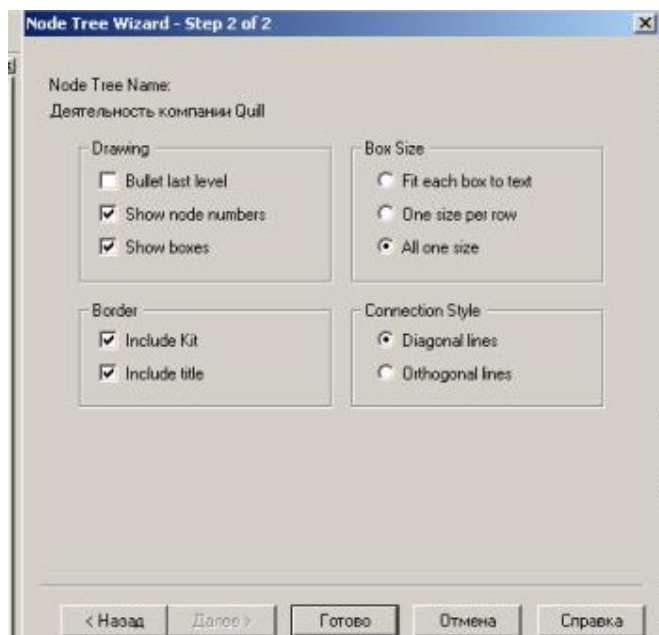
**Рис. 27.** Вибір команди для формування діаграми дерева вузлів

Для створення *діаграми дерева вузлів* слід вибрати в меню пункт Diagram/Add Node Tree (рис. 27). Виникає діалог формування *діаграми дерева вузлів* Node Tree Definition (рис. 28,29).



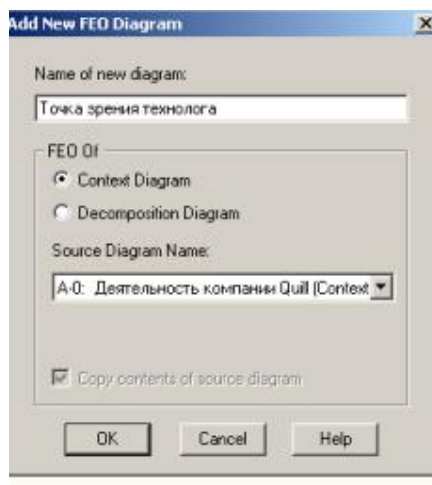
**Рис. 28.** Діалог настроювання діаграми дерева вузлів (крок 1)

У діалозі Node Tree Definition слід указати глибину дерева – Number of Levels ( за замовчуванням - 3) і *корінь дерева* ( за замовчуванням - батьківська *робота* поточної діаграми). За замовчуванням нижній рівень декомпозиції показується у вигляді списку, інші *роботи* - у вигляді прямокутників. Для відображення всього дерева у вигляді прямокутників слід виключити опцію *Bullet Last Level*. При створенні дерева вузлів слід указати ім'я діаграми, оскільки, якщо в декількох діаграмах у якості кореня на дереві вузлів використовувати ту саму *роботу*, усі ці діаграми отримають однаковий номер (номер вузла + постфікс N, наприклад A0N) і в списку відкритих діаграм (пункт меню Window) їх можна буде розрізнити лише за іменем.



**Рис. 29.** Діалог настроювання діаграми дерева вузлів (крок 2)

Діаграми "тільки для експозиції" (*FEO*) часто використовуються в моделі для ілюстрації інших точок зору, для відображення окремих деталей, які не підтримуються явно синтаксисом IDEF0. Діаграми *FEO* дозволяють порушити будь-яке синтаксичне правило, оскільки по суті є просто картинками – копіями стандартних діаграм і не включаються в аналіз синтаксису. Для створення діаграми *FEO* слід вибрати пункт меню Diagram/Add *FEO* Diagram. У виникаючому діалозі Add New *FEO* Diagram слід указати ім'я діаграми *FEO* і тип батьківської діаграми (рис. 30).

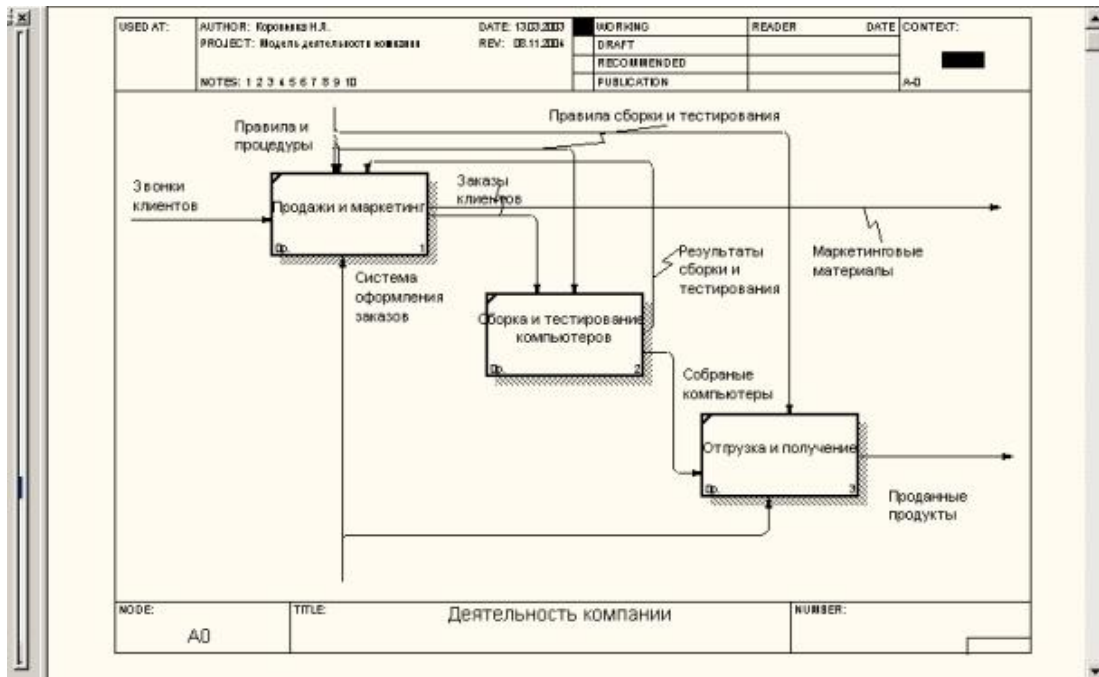


**Рис. 30.** Діалог створення *FEO*-Діаграми

Нова діаграма отримує номер, який генерується автоматично (номер батьківської діаграми по вузлу + постфікс F, наприклад A1F ).

## Каркас діаграми

На рис. 31 показано типовий приклад *діаграми декомпозиції* з граничними рамками, які називаються каркасом діаграми.

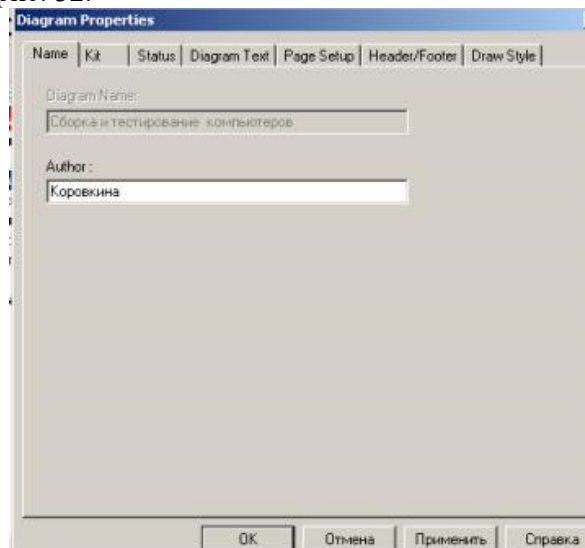


**Рис. 31.** Приклад *діаграми декомпозиції* з каркасом

Каркас містить заголовок (верхня частина рамки) і підвал (нижня частина). Заголовок каркаса використовується для відстеження *діаграми* в процесі моделювання. Нижня частина використовується для ідентифікації й позиціонування в ієрархії *діаграми*.

Зміст елементів каркаса наведений у табл. 1 і 2.

Значення полів каркаса задаються в діалозі Diagram Properties (меню Diagram /Diagram Properties) - рис. 32.



**Рис. 32.** Діалог *Diagram Properties*

Таблиця 7.1.

Поля заголовка каркаса (ліворуч праворуч)

Поле	Зміст
Used At	Використовується для вказівки на батьківську <i>роботу</i> у випадку, якщо на поточну діаграму посилалися за допомогою <i>стрілки</i> виклику
Autor, Date, Rev, Project	Ім'я автора діаграми, дата створення й ім'я проекту, в межах якого була створена діаграма. Rev- <i>Дата</i> останнього редагування діаграми
Notes 12345678910	Використовується при проведенні сеансу експертизи. Експерт повинен (на паперовій копії діаграми) указати число зауважень, викреслюючи цифру зі списку щораз при внесенні нового зауваження
Status	Статус відображає стадію створення діаграми, відображаючи всі етапи публікації
Working	Нова діаграма, кардинально оновлена діаграма або новий автор діаграми
Draft	Діаграма пройшла первинну експертизу й готова до подальшого обговорення
Recommended	Діаграма й усі її супровідні документи <b>пройшли</b> експертизу. Нових змін не очікується
Publication	Діаграма готова до остаточного друку й публікації
Reader	Ім'я читача (експерта)
Date	Дата прочитання (експертизи)
Context	Схема розташування <i>робіт</i> у діаграмі верхнього рівня. <i>Робота</i> , що є батьківською, показана темним прямокутником, інші - світлим. На <i>контекстній діаграмі</i> (A-0) показаний напис TOP. У лівому нижньому куті показується номер по вузлу батьківської діаграми: <div data-bbox="485 1730 854 1839" data-label="Diagram"> </div>

## Злиття й розщеплення моделей

Можливість злиття й розщеплення моделей забезпечує колективну *роботу* над проектом. Так, *керівник проекту* може створити декомпозицію верхнього рівня й дати завдання аналітикам продовжити декомпозицію кожної гілки дерева у вигляді окремих моделей. Після закінчення *роботи* над окремими гілками всі підмоделі можуть бути злиті в єдину модель. З іншого боку, окрема *гілка* моделі може бути відділена для використання як незалежної моделі, для доробки або архівування.

Таблиця 7.2. Поля підвалу каркаса

Поле	Зміст
Node	Номер вузла діаграми (номер батьківської <i>роботи</i> )
Title	Ім'я діаграми. За замовчуванням - ім'я батьківської <i>роботи</i>
Number	C-Number, унікальний номер версії діаграми
Page	Номер сторінки, може використовуватися як номер сторінки при формуванні папки

AllFusion Process Modeler використовує для злиття й розгалуження моделей *стрілки* виклику.

Для злиття необхідно виконати наступні умови:

- обидві моделі, що зливаються, повинні бути відкриті в AllFusion Process Modeler.
- ім'я моделі-джерела, яке приєднують до моделі-мети, повинне збігатися з іменем *стрілки* виклику *роботи* в моделі-меті.
- *стрілка* виклику повинна виходити з *роботи*, що не піддавалась декомпозиції (*робота* повинна мати діагональну рису в лівому верхньому куті) (рис. 33).
- імена контекстної *роботи* моделі-джерела, що приєднується й *роботи* на моделі-меті, до якої ми приєднуємо модель-джерело, повинні збігатися.
- модель-джерело повинна мати, принаймні, одну *діаграму декомпозиції*.



Рис. 33. Стрілка виклику роботи "Складання й тестування комп'ютерів" моделі-мети

Для злиття моделей потрібно клацнути правою кнопкою миші по роботі зі *стрілкою* виклику в моделі-мети й у спливаючому меню вибрати пункт *Merge Model*.

З'являється діалог, у якому слід указати опції злиття моделі (рис. 34). При злитті моделей поєднуються й словники стрілок і *робіт*. У випадку однакових визначень можливий перезапис визначень або прийняття визначень із моделі-джерела. Те саме відбувається із іменами стрілок, сховищам даних і зовнішнім посиланням. (Сховища даних і зовнішні посилання - об'єкти *діаграм потоків даних, DFD*, будуть розглянуті у наступній лабораторній роботі.)

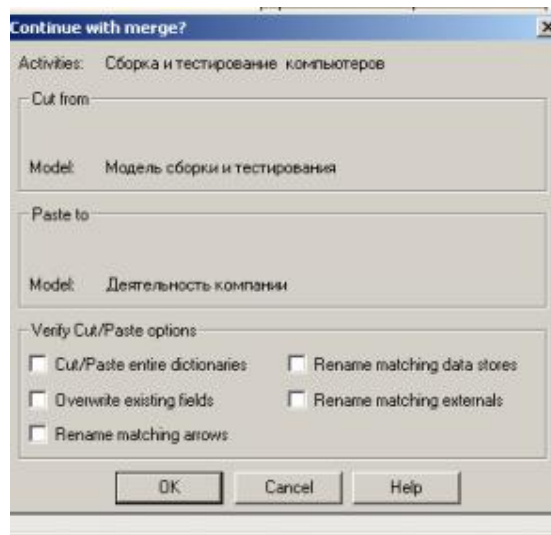


Рис. 34. Діалог Continue with merge

Після підтвердження злиття (кнопка ОК) модель-джерело приєднується до моделі-мети, *стрілка* виклику зникає, а *робота*, від якої відходила *стрілка* виклику, стає декомпозованою – до неї приєднується *діаграма* декомпозиції першого рівня моделі-джерела. *Стрілки*, що стосуються *роботи* на діаграмі моделі-мети, автоматично не мігрують у декомпозицію, а відображаються як недозволені. Їх необхідно тунелювати вручну.

У процесі злиття модель-джерело залишається незмінною, і до моделі-мети підключається фактично її копія. Не потрібно плутати злиття моделей із синхронізацією. Якщо надалі модель-джерело буде редагуватися, ці зміни автоматично не потраплять у відповідну *гілку* моделі-мети.

Поділ моделей проводиться аналогічно. Для розгалуження моделі слід клацнути правою кнопкою миші по *роботі*, що піддавалась *декомпозиції* (*робота* не повинна мати діагональної риси в лівому верхньому куті) і вибрати в випливаючому меню пункт *Split Model*. У діалозі, що з'явився, Split Options слід указати ім'я створюваної моделі. Після підтвердження розщеплення в старій моделі *робота* стане недекомпонованою (ознака – діагональна риса в лівому верхньому куті), буде створена *стрілка* виклику, її ім'я буде збігатися з іменем нової моделі, і, нарешті, буде створена нова модель, причому ім'я контекстної *роботи* буде збігатися з іменем *роботи*, від якої була "відірвана" *декомпозиція*.

### 3. Порядок виконання роботи

- 3.1. Ознайомитися з теоретичною частиною.
- 3.2. Ознайомитися із середовищем моделювання AllFusion Process Modeler (BPwin).
- 3.3. Обрати завдання згідно порядкового номеру у журналі зі списку індивідуальних завдань (за згодою викладача можливо запропонувати власну тему для системи, що моделюється).
- 3.4. Розробити функціональну модель (IDEFO-модель) інформаційної системи для свого індивідуального завдання.
- 3.5. Оформити звіт по результатах виконаної роботи як «Лабораторна робота №1».
- 3.6. Розробити діаграми дерева вузлів і FEO для свого індивідуального завдання.
- 3.7. Оформити звіт по результатах виконаної роботи як «Лабораторна робота №2».

#### Вимоги до звіту

Оформити звіт для захисту лабораторної роботи за зразком:

- назва роботи;
- мета роботи;
- короткі теоретичні відомості;
- порядок роботи (у цьому пункті бажано максимально детально відобразити процес створення Вами необхідних моделей та діаграм, навести покрокові скріншоти із поясненнями);
- аналіз отриманих результатів та висновок.

У звіті обов'язково повинні бути наведені:

- ✓ Контекстна діаграма
- ✓ Діаграма декомпозиції 1-го рівня
- ✓ Дві діаграми декомпозиції 2-го рівня для 2-ох найбільш цікавих блоків з діаграми декомпозиції 1-го рівня
- ✓ Діаграма дерева вузлів



### Рекомендована література

1. . Шаховська Н. Б., Литвин В. В. Проектування інформаційних систем: навчальний посібник / Н. Б. Шаховська, В. В. Литвин. -Львів: "Магнолія-2006", 2011. - 380 с.
2. Вендров А. М. Проектирование программного обеспечения экономических информационных систем : учебник / А. М. Вендров. – 2-е изд., перераб. и доп. – М. : Финансы и статистика, 2006. – 544 с.
3. Гвоздева В. А. Основы построения автоматизированных информационных систем : учебник / В. А. Гвоздева, Ю. И. Лаврентьева. – М. : ИД "Форум": ИНФРА-М, 2007. – 320 с.
4. Грекул В. И. Проектирование информационных систем : учебн. пособ. / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – М. : БИНОМ. Лаборатория знаний, 2008. – 300 с.
5. Избачков Ю. С. Информационные системы : учебник / Ю. С. Избачков, В. Н. Петров. – 2-е изд. – СПб. : Питер, 2006. – 656 с.
6. Дубейковский В. И. Практика функционального моделирования с AllFusion Process Modeler 4.1. (ВРwin) Где? Зачем? Как? / В.И. Дубейковский. – М. : ДИАЛОГ-МИФИ, 2004. – 464 с

Методичні вказівки до лабораторних робіт № 1-2 «Моделювання бізнес-процесів у середовищі AllFusion® Process Modeler» з дисципліни **“Проектування інформаційних систем”** для студентів спеціальності 122 “Комп’ютерні науки та інформаційні технології”

Укладач:

**Дорошенко А.В.**

