

## 4. Структурний аналіз бізнес-процесів

### 4.1. Діаграми структурного системного аналізу

Під структурним системним аналізом (ССА) прийнято розуміти метод дослідження системи, який починається з найбільш загального її опису з подальшою деталізацією представлення окремих аспектів її поведінки і функціонування. При цьому загальна модель системи будується у вигляді певної ієрархічної структури, яка відображає різні рівні абстракції з обмеженим числом компонентів на кожному з рівнів. Одним із головних принципів структурного системного аналізу є виділення на кожному з рівнів абстракції тільки найбільш істотних компонентів або елементів системи [17].

У рамках цього напрямку програмної інженерії прийнято розглядати три графічні нотації, що отримали назви діаграм: *діаграми «сутність-зв'язок»* (Entity-Relationship Diagrams, ERD), *діаграми функціонального моделювання* (Structured Analysis and Design Technique, SADT) і *діаграми потоків даних* (Data Flow Diagrams, DFD).

#### 1. Діаграми «сутність-зв'язок»

Ця нотація була запропонована П. Ченом (P. Chen) у його відомій праці 1976 року і отримала подальший розвиток у роботах Р. Баркера (R. Barker). Діаграми «сутність-зв'язок (ERD)» призначені для графічного представлення моделей даних програмної системи, що розробляється, і пропонують певний набір стандартних позначень для визначення даних і відносин між ними. За допомогою цього виду діаграм можна описати окремі компоненти концептуальної моделі даних і сукупність взаємозв'язків між ними, що мають важливе значення для системи, що розробляється.

Основними поняттями цієї нотації є поняття сутності і зв'язку. При цьому під сутністю (entity) розуміється довільна множина реальних або абстрактних об'єктів, кожний з яких володіє однаковими властивостями і характеристиками. В цьому випадку кожен об'єкт може бути *екземпляром*

однієї і лише однієї сутності, повинен мати унікальне ім'я або ідентифікатор, а також відрізнятися від інших екземплярів певної сутності.

Прикладами сутностей можуть бути: банк, клієнт банку, рахунок клієнта, аеропорт, пасажир, рейс, комп'ютер, термінал, автомобіль, водій. Кожна із сутностей може розглядатися з різним ступенем деталізації і на різному рівні абстракції, що визначається конкретною постановкою завдання. Для графічного представлення сутності використовуються спеціальні позначення (рис. 4.1).



Рис. 4.1. Графічні зображення для позначення сутності

*Зв'язок* (relationship) визначається як співвідношення або деяка асоціація окремих сутностей. Прикладами зв'язків можуть бути споріднені відносини типу «батько-син» або виробничі відносини типу «начальник-підлеглий». Інший тип зв'язків задається відносинами «мати у власності» або «володіти властивістю». Різні типи зв'язків графічно зображуються у формі ромба з відповідним ім'ям певного зв'язку (рис. 4.2).

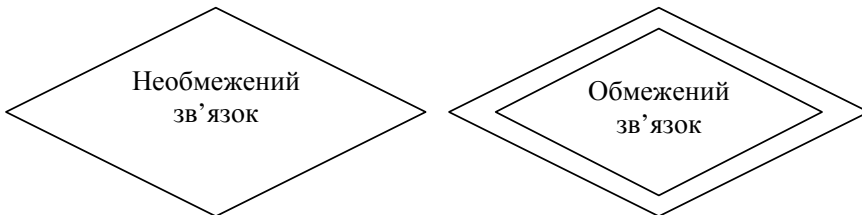


Рис. 4.2. Графічні зображення для позначення зв'язків

Графічна модель даних будується так, щоб зв'язки між окремою сутністю відображали не тільки семантичний характер відповідного співвідношення, але й додаткові аспекти обов'язковості зв'язків, а також кратність сутності, що бере участь у таких відносинах екземплярів.

Розглянемо як простий приклад ситуацію, яка описується двома сутностями: «Співробітник» і «Компанія». При цьому як зв'язок зазвичай використовують приналежність співробітника до даної компанії. Якщо врахувати міркування про те, що в компанії працюють декілька співробітників, і ці співробітники не можуть бути працівниками інших компаній, то ця інформація може бути представлена графічно у вигляді наступної діаграми «сутність-зв'язок» (рис. 4.3). На цьому рис. літера «N» біля зв'язку означає

той факт, що в компанії можуть працювати більш за одного співробітника, при цьому значення N наперед не фіксується. Цифра «1» на іншому кінці зв'язку означає, що співробітник може працювати тільки в одній конкретній компанії, тобто не допускається прийом на роботу співробітників за сумісництвом з інших компаній або установ.



Рис. 4.3. Діаграма «сутність-зв'язок» для прикладу співробітників деякої компанії

Дещо інша картина складається у ході розгляду сутності «співробітник» і «проект» і зв'язку «бере участь в роботі над проектом» (рис. 4.4). Оскільки загалом один співробітник може брати участь у розробці декількох проектів, а в розробці одного проекту можуть брати участь декілька співробітників, то такий зв'язок є багатозначним. Цей факт спеціально відображено на діаграмі літерами «N» і «M» біля відповідної сутності, при цьому вибір конкретних літер не є принциповим.



Рис. 4.4. Діаграма «сутність-зв'язок» для прикладу співробітників, що беруть участь у роботі над проектами

Розглянуті дві діаграми можуть бути об'єднані в одну, на якій буде представлена інформація про співробітників компанії, що беруть участь у розробці проектів певної компанії (рис. 4.5). При цьому може бути введений додатковий зв'язок, що характеризує проекти певної компанії.

На вказаних діаграмах можуть бути відображені складніші залежності між окремою сутністю, яка відображає обов'язковість виконання деяких додаткових умов, визначених специфікою вирішуваного завдання і модельованої предметної області. Зокрема, можуть бути відображені зв'язки підпорядкування однієї сутності іншій або введені обмеження на дію окремих зв'язків. У подібних випадках використовуються додаткові графічні позначення, що відображають особливості відповідної семантики.

Обмеженість ERD виявляється при конкретизації концептуальної моделі в детальніше представлення модельованої програмної системи, яке, окрім статичних зв'язків, повинне вмещувати інформацію про поведінку або функціонування окремих її компонентів. З цією метою в рамках ССА використовується інший тип діаграм, що отримали назву діаграм потоків даних.

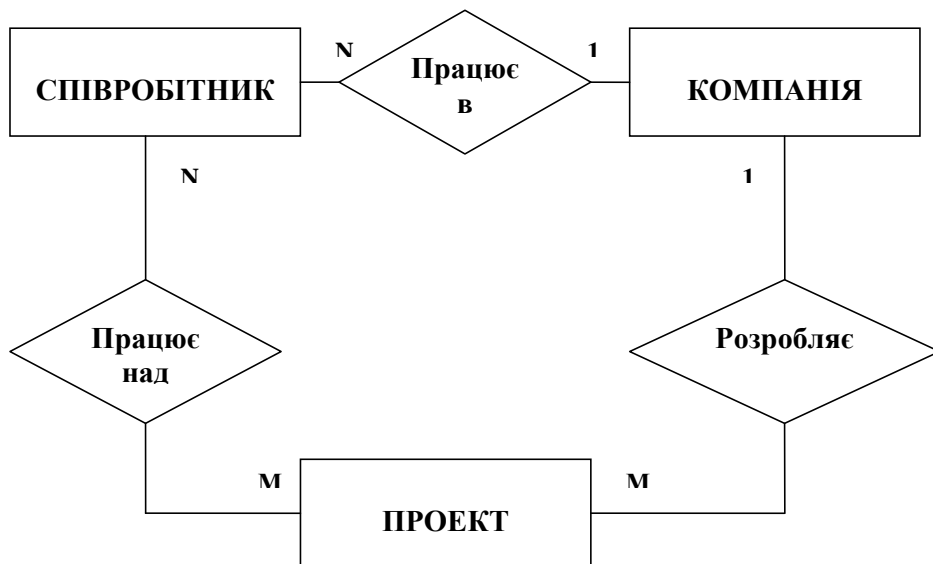


Рис. 4.5. Діаграма «сутність-зв'язок» для загального прикладу компанії

## 2. Діаграми функціонального моделювання

Початок розробки діаграм функціонального моделювання припадає на середину 1960-х років, коли Дуглас Т. Росс запропонував спеціальну техніку моделювання, що отримала назву SADT (Structured Analysis & Design Technique). Військово-повітряні сили США використовували методику SADT як частину своєї програми *інтеграції комп'ютерних і промислових технологій* (Integrated Computer Aided Manufacturing, ICAM) і назвали її IDEF (Icam DEFinition). Метою програми ICAM було збільшення ефективності комп'ютерних технологій у сфері проектування нових засобів озброєнь і ведення бойових дій. Одним із результатів цих досліджень був висновок про те, що описові мови не ефективні для документування і моделювання процесів функціонування складних систем. Подібні описи на природній мові не забезпечують необхідного рівня несуперечності і повноти, що мають домінуюче значення при розв'язанні задач моделювання.

У рамках програми ICAM було розроблено декілька графічних мов моделювання, які отримали наступні назви [15,16]:

- Нотація IDEF0 – для документування процесів виробництва і відображення інформації про використання ресурсів на кожному з етапів проектування систем.
- Нотація IDEF1 – для документування інформації про виробниче оточення систем.
- Нотація IDEF2 – для документування поведінки системи в часі.
- Нотація IDEF3 – спеціально для моделювання бізнес-процесів.

Нотація IDEF2 ніколи не була повністю реалізована. Нотація IDEF1 у

1985 році була розширена і перейменована в IDEF1X. Методологія IDEF-SADT знайшла застосування в урядових і комерційних організаціях, оскільки на той період часу цілком задовольняла різним вимогам, що висуваються до моделювання широкого класу систем.

На початку 1990 року спеціально створена група користувачів IDEF (IDEF Users Group), разом із Національним інститутом зі стандартизації і технології США (National Institutes for Standards and Technology, NIST), здійснила спробу створення стандарту для IDEF0 і IDEF1X. Ця спроба виявилася успішною і завершилася ухваленням у 1993 році стандарту уряду США, відомого як FIPS для даних двох технологій IDEF0 і IDEF1X. Протягом подальших років цей стандарт продовжував активно розвиватися і став основою для реалізації в деяких перших CASE-засобах.

Методологія IDEF-SADT є сукупністю методів, правил і процедур, призначених для побудови функціональної моделі системи будь-якої предметної області. Функціональна модель SADT відображає структуру процесів функціонування системи і її окремих підсистем, тобто виконуваними ними дії і зв'язки між цими діями. З цією метою будуються спеціальні моделі, які дозволяють наочно представити послідовність певних дій. Початковими будівельними блоками будь-якої моделі IDEF0 процесу є діяльність (activity) і стрілки (arrows).

Розглянемо стисло основні поняття методології IDEF-SADT, які використовуються при побудові діаграм функціонального моделювання. *Діяльність* є певна дія або набір дій, які мають фіксовану мету і приводять до деякого кінцевого результату. Іноді діяльність називають просто процесом. Моделі IDEF0 відстежують різні види діяльності системи, їх опис і взаємодію з іншими процесами. На діаграмах діяльність або процес зображається прямокутником, який називається блоком. *Стрілка* слугує для позначення певного носія або дії, які забезпечують перенесення даних або об'єктів від однієї діяльності до іншої. Стрілки також необхідні для опису того, хто саме проводить діяльність, і які ресурси вона споживає. Це так звані ролі стрілок – ICOM – скорочення перших літер від назв відповідних стрілок IDEF0. При цьому розрізняють стрілки чотирьох видів:

- I (Input) – вхід, тобто все, що надходить у процес або споживається процесом.
- C (Control) – управління або обмеження на виконання операцій процесу.
- O (Output) – вихід або результат процесу.
- M (Mechanism) – механізм, який використовується для виконання процесу.

Методологія IDEF0 однозначно визначає, яким чином зображуються на діаграмах стрілки кожного виду ICOM. Стрілка Вхід (Input) виходить

з лівого боку рамки робочого поля і входить зліва в прямокутник процесу. Стрілка Управління (Control) входить і виходить згори. Стрілка Вихід (Output) виходить з правого боку процесу і входить у правий бік рамки. Стрілка Механізм (Mechanism) входить у прямокутник процесу знизу. Таким чином, базове представлення процесу на діаграмах IDEF0 має наступний вигляд (рис. 4.6).

Техніка побудови діаграм є головною особливістю методології IDEF-SADT. Місце з'єднання стрілки з блоком визначає тип інтерфейсу. При цьому всі функції модельованої системи та інтерфейси на діаграмах представляються у вигляді відповідних блоків процесів і стрілок ICOM. Інформація з управління входить у блок згори, тоді як інформація, яка піддається обробці, зображується з лівого боку блоку. Результати процесу представляються як виходи процесу і показуються з правого боку блоку. Як механізм може виступати людина або автоматизована система, які реалізують цю операцію. Відповідний механізм на діаграмі представляється стрілкою, яка входить у блок процесу знизу.

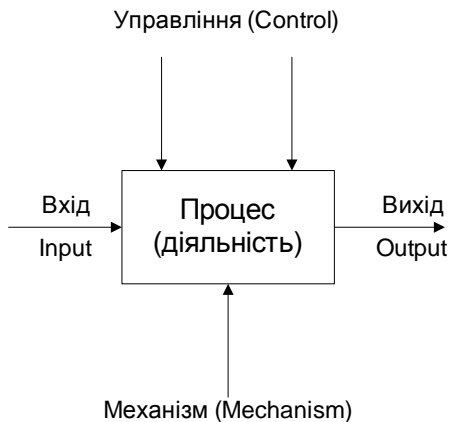


Рис. 4.6. Позначення процесу і стрілок ICOM на діаграмах IDEF0

Однією з найбільш важливих особливостей методології IDEF-SADT є поступове введення все більш детальних представлень моделі системи в процесі розробки окремих діаграм. Побудова моделі IDEF-SADT починається з представлення всієї системи у вигляді простої діаграми, що складається з одного блоку процесу і стрілок ICOM, які слугують для зображення основних видів взаємодії з об'єктами поза системою. Оскільки початковий процес представляє всю систему як єдине ціле, таке уявлення є найбільш загальним і підлягає подальшій декомпозиції.

Для ілюстрації основних ідей методології IDEF-SADT розглянемо наступний приклад. Як процес представимо діяльність з оформлення кредиту в банку. Входом цього процесу є заявка від клієнта на отримання

кредиту, а виходом – відповідний результат, тобто безпосередньо кредит. При цьому інформацією з управління є правила оформлення кредиту, які регламентують умови отримання відповідних фінансових коштів у кредит. Виконавцем такого процесу є службовець банку, який уповноважений виконати всі операції з оформлення кредиту. Приклад початкового уявлення процесу показаний на рис. 4.7.

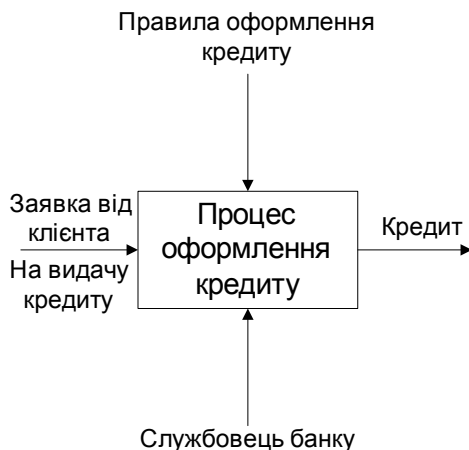


Рис. 4.7. Приклад початкової діаграми IDEF-SADT для процесу оформлення кредиту в банку

Зрештою, модель IDEF-SADT є серією ієрархічно взаємопов'язаних діаграм із супровідною документацією, яка розбиває початкове представлення складної системи на окремі складові частини. Деталі кожного з основних процесів представляються у вигляді детальніших процесів на інших діаграмах. У цьому випадку кожна діаграма нижнього рівня є декомпозицією деякого процесу з більш загальної діаграми. Тому на кожному кроці декомпозиції більш загальна діаграма конкретизується в низці детальніших діаграм.

Нині діаграми структурного системного аналізу IDEF-SADT продовжують використовуватися багатьма організаціями для побудови і детального аналізу функціональної моделі існуючих на підприємстві бізнес-процесів, а також для розробки нових бізнес-процесів. Основний недолік цієї методології пов'язаний із відсутністю явних засобів для об'єктно-орієнтованого представлення моделей складних систем. Хоча деякі аналітики відзначають важливість знання і застосування нотації IDEF-SADT, обмежені можливості цієї методології стосовно реалізації відповідних графічних моделей в об'єктно-орієнтованому програмному коді істотно звужують діапазон розв'язуваних за її допомогою задач.

Зараз до родини IDEF можна віднести наступні стандарти:

- IDEF0 – методологія функціонального моделювання. За допомо-

гою наочної графічної мови IDEF0 система, що вивчається, постає перед розробниками й аналітиками у вигляді набору взаємопов'язаних функцій (функціональних блоків – у термінах IDEF0). Як правило, моделювання засобами IDEF0 є першим етапом вивчення будь-якої системи;

- IDEF1 – методологія моделювання інформаційних потоків усередині системи, що дозволяє відображати і аналізувати їх структуру і взаємозв'язки;

- IDEF1X (IDEF1 Extended) – методологія побудови реляційних структур. IDEF1X відноситься до типу методологій «Сутність-взаємозв'язок (ER – Entity-Relationship)» і, як правило, використовується для моделювання реляційних баз даних, що стосуються такої системи;

- IDEF2 – методологія динамічного моделювання розвитку систем. У зв'язку з вельми серйозними складнощами аналізу динамічних систем від цього стандарту практично відмовилися, і його розвиток припинився на самому початковому етапі. Проте нині присутні алгоритми та їх комп'ютерні реалізації, що дозволяють перетворювати набір статичних діаграм IDEF0 на динамічні моделі, побудовані на базі «розфарбованих мереж Петрі» (CPN – Color Petri Nets);

- IDEF3 – методологія документування процесів, що відбуваються в системі, яка використовується, наприклад, при дослідженні технологічних процесів на підприємствах. За допомогою IDEF3 описуються сценарій і послідовність операцій для кожного процесу. IDEF3 має прямий взаємозв'язок з методологією IDEF0 – кожна функція (функціональний блок) може бути представлена у вигляді окремого процесу засобами IDEF3;

- IDEF4 – методологія побудови об'єктно-орієнтованих систем. Засоби IDEF4 дозволяють наочно відображати структуру об'єктів і закладені принципи їх взаємодії, тим самим дозволяючи аналізувати і оптимізувати складні об'єктно-орієнтовані системи;

- IDEF5 – методологія онтологічного дослідження складних систем. За допомогою методології IDEF5 онтологія системи може бути описана за допомогою певного словника термінів і правил, на підставі яких можуть бути сформовані достовірні твердження про стан цієї системи в певний момент часу. На основі цих тверджень формуються висновки про подальший розвиток системи і проводиться її оптимізація.

### **3. Діаграми потоків даних**

Основою цієї методології графічного моделювання інформаційних систем є спеціальна технологія побудови діаграм потоків даних DFD. У розробці методології DFD взяли участь багато аналітиків, серед яких слід зазначити Е. Йордона (E. Yourdon). Він є автором однієї з перших графічних нотацій DFD. На сьогодні найбільш поширеною є так звана нотація Гейна-Сарсона (Gene-Sarson), основні елементи якої будуть розглянуті в цьому розділі.



Модель системи в контексті DFD представляється у вигляді деякої інформаційної моделі, основними компонентами якої є різні потоки даних, що переносять інформацію від однієї підсистеми до іншої. Кожна з підсистем виконує певні перетворення вхідного потоку даних і передає результати обробки інформації у вигляді потоків даних для інших підсистем [15,16].

Основними компонентами діаграм потоків даних є:

- зовнішня сутність;
- накопичувачі даних або сховища;
- процеси;
- потоки даних;
- системи/підсистеми.

*Зовнішня сутність* є матеріальним об'єктом або фізичною особою, які можуть виступати як джерело або приймач інформації. Визначення певного об'єкта або системи як зовнішня сутність не є строго фіксованим. Хоча зовнішня сутність знаходиться за межами певної системи, в процесі подальшого аналізу деяка зовнішня сутність може бути перенесена всередину діаграми моделі системи. З іншого боку, окремі процеси можуть бути винесені за межі діаграми і представлені як зовнішня сутність. Прикладами зовнішньої сутності можуть бути: клієнти організації, замовники, персонал, постачальники.

Зовнішня сутність позначається прямокутником з тінню (рис. 4.8), усередині якого вказується її ім'я. При цьому як ім'я рекомендується використовувати іменник в називному відмінку. Іноді зовнішню сутність називають також термінатором.

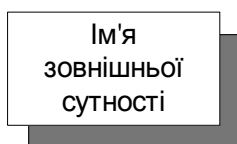


Рис. 4.8. Зображення зовнішньої сутності на діаграмі потоків даних

*Процес* є сукупністю операцій щодо перетворення вхідних потоків даних на вихідні відповідно до певного алгоритму або правила. Хоча фізично процес може бути реалізований різними способами, найчастіше йдеться про програмну реалізацію процесу. Процес на діаграмі потоків даних зображується прямокутником із закругленими вершинами (рис. 4.9), розділеним на три секції, або поля горизонтальними лініями. Поле номера процесу слугує для ідентифікації останнього. В середньому полі вказується ім'я процесу. Як ім'я рекомендовано використовувати дієслово в невизначеній формі з необхідними доповненнями. Нижнє поле містить вказівку на спосіб фізичної реалізації процесу.

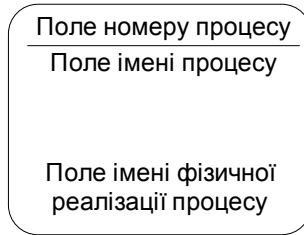


Рис. 4.9. Зображення процесу на діаграмі потоків даних

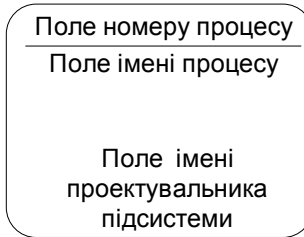


Рис. 4.10. Зображення підсистеми на діаграмі потоків даних

Інформаційна модель системи будується як деяка ієрархічна схема у вигляді так званої контекстної діаграми, на якій початкова модель послідовно представляється у вигляді моделі підсистем відповідних процесів перетворення даних. При цьому *підсистема* або система на контекстній діаграмі DFD зображується так само, як і процес – прямокутником із закругленими вершинами (рис. 4.10).

*Накопичувач даних*, або сховище – абстрактний пристрій або спосіб зберігання інформації, що передається між процесами. Передбачається, що дані можна у будь-який момент помістити в накопичувач і через деякий час витягнути, причому фізичні способи приміщення і витягання даних можуть бути довільними. Накопичувач даних може бути фізично реалізований у різні способи, але найчастіше передбачається його реалізація в електронному вигляді на магнітних носіях. Накопичувач даних на діаграмі потоків даних зображується прямокутником з двома полями (рис. 4.11). Перше поле слугує для вказівки номера або ідентифікатора накопичувача, який починається з літери «D». Друге поле слугує для вказівки імені. При цьому як ім'я накопичувача рекомендується використовувати іменник, який характеризує спосіб зберігання відповідної інформації.

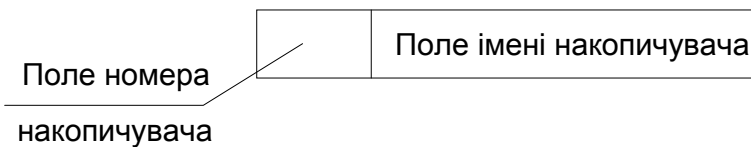


Рис. 4.11. Зображення накопичувача на діаграмі потоків даних

Нарешті, *потік даних* визначає якісний характер інформації, що передається через певне з'єднання від джерела до приймача. Реальний потік даних може передаватися по мережі між двома комп'ютерами або будь-яким іншим способом, що допускає витягання даних та їх відновлення в необхідному форматі. Потік даних на діаграмі DFD зображується лінією зі стрілкою на одному з її кінців, при цьому стрілка показує напрям потоку даних. Кожен потік даних має своє власне ім'я, зміст, що відображає його.

Таким чином, інформаційна модель системи в нотації DFD будується у вигляді діаграм потоків даних, які графічно представляються з використанням відповідної системи позначень. Як приклад розглянемо спрощену модель процесу отримання певної суми готівкою по кредитній картці клієнтом банку. Зовнішньою сутністю цього прикладу є клієнт банку і, можливо, службовець банку, який контролює процес обслуговування клієнтів. Накопичувачем даних може бути база даних про стан рахунків окремих клієнтів банку. Окремі потоки даних відображають характер інформації, що передається, необхідної для обслуговування клієнта банку. Відповідна модель для такого прикладу може бути представлена у вигляді діаграми потоків даних (рис. 4.12).

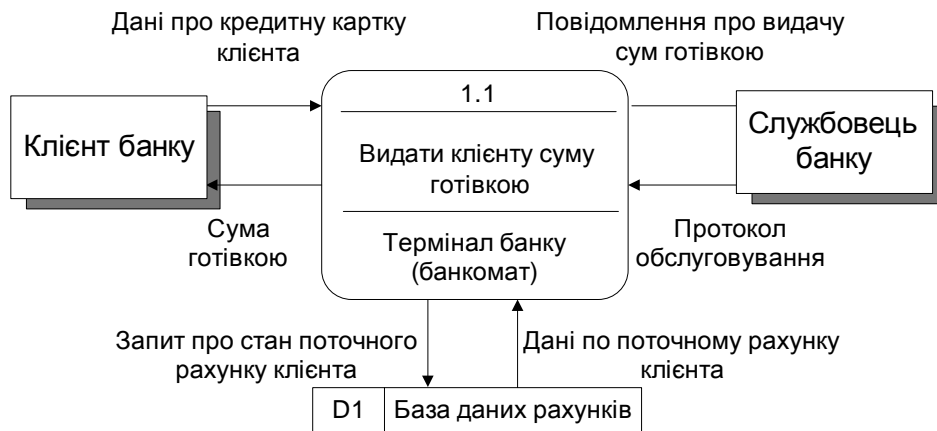


Рис. 4.12. Приклад діаграми DFD для процесу отримання певної суми готівкою по кредитній картці

На сьогодні діаграми потоків даних використовуються в деяких CASE-засобах для побудови інформаційних моделей систем обробки даних. Основний недолік цієї методології також пов'язаний із відсутністю явних засобів для об'єктно-орієнтованого представлення моделей складних систем, а також для представлення складних алгоритмів обробки даних. Оскільки на діаграмах DFD не вказуються характеристики часу виконання окремих процесів і передачі даних між процесами, то моделі систем, що реалізують синхронну обробку даних, не можуть бути адекватно представлені в

нотації DFD. Усі ці особливості методології структурного системного аналізу обмежили можливості її широкого застосування і послужили основою для включення відповідних засобів в уніфіковану мову моделювання.

## **4.2. Функціональний підхід до моделювання бізнес-процесів**

### **4.2.1. Історія виникнення стандарту IDEF0**

Методологію IDEF0 можна вважати наступним етапом розвитку добре відомої графічної мови опису функціональних систем SADT (Structured Analysis and Design Technique). Історично IDEF0 як стандарт був розроблений у 1981 році в рамках широкої програми автоматизації промислових підприємств, яка мала позначення ICAM (Integrated Computer Aided Manufacturing) і була запропонована департаментом Військово-повітряних Сил США. Власне родина стандартів IDEF успадкувала своє позначення від назви цієї програми (IDEF=ICAM DEFinition). У процесі практичної реалізації учасники програми ICAM зіткнулися з необхідністю розробки нових методів аналізу процесів взаємодії в промислових системах. При цьому, окрім вдосконаленого набору функцій для опису бізнес-процесів, однією з вимог до нового стандарту була наявність ефективної методології взаємодії в рамках «аналітик-фахівець». Іншими словами, новий метод повинен був забезпечити групову роботу над створенням моделі з безпосередньою участю всіх аналітиків і фахівців, зайнятих у рамках проекту [28].

У результаті пошуку відповідних рішень народилася методологія функціонального моделювання IDEF0. З 1981 року стандарт IDEF0 зазнав декількох незначних змін, в основному обмежувального характеру, і остання його редакція була підготовлена у грудні 1993 року Національним Інститутом зі Стандартів і Технологій США (NIST).

### **4.2.2. Методика IDEF0**

Основним об'єктом діаграми процесів в нотації IDEF0 є об'єкт Activity. Графічно він є чотирикутником, що зображає функції, які виконуються в організації (рис. 4.13).

Нагадаємо, що кожен функцію (процедуру, роботу) можна розглядати як певний процес. На верхньому рівні кожен процес може бути розглянутий як «чорний ящик», що перетворює вхідні ресурси на вихідні. Таке визначення фактично збігається з визначенням процесу в MS ISO 9000:2000 [2].

Другою основною складовою стандарту IDEF0 є стрілки (див. рис. 4.13). На діаграмі процесу в IDEF0 стрілки, що входять у функцію зліва,

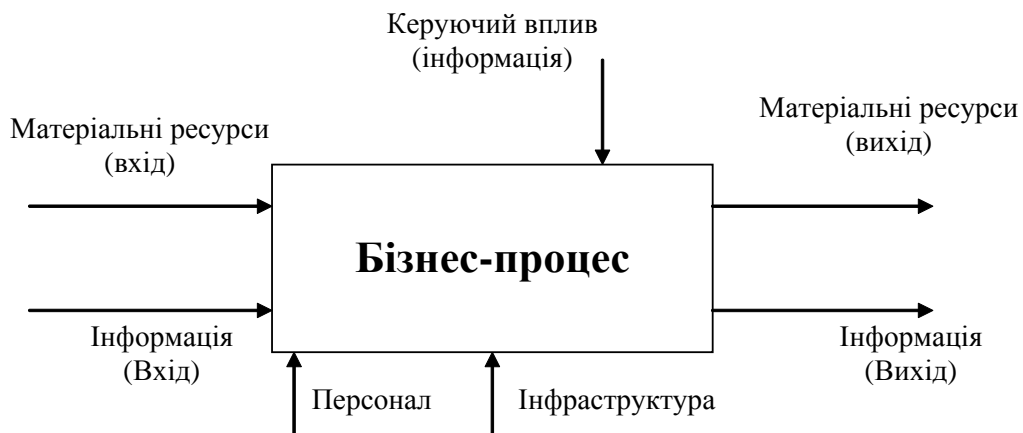


Рис. 4.13. Формування моделі бізнес-процесу. Крок 1

служать для опису потоків матеріальних ресурсів або потоків інформації, документів.

Вхідні ресурси перетворюються функцією (роботою, процесом). Результатом цього перетворення є матеріальні виходи або інформація, які зображено у вигляді стрілок, що виходять з правої сторони чотирикутника.

Для виконання будь-якої реальної роботи необхідні основні засоби, інструменти, персонал, програмні продукти тощо. Всі ці ресурси відображаються на діаграмі стрілками, що входять у чотирикутник знизу.

Що ще необхідно показати на діаграмі для того, щоб можна було описати реальний процес організації? Слід відобразити дії, які визначають порядок виконання роботи, управляють роботою. Такими діями можуть бути, наприклад, усне розпорядження керівника, нормативний документ, державний, галузевий стандарт, технічні умови тощо. Дії з управління показано на діаграмі стрілками згори. Будь-яка дія з управління існує у вигляді певної інформації, тому стрілки згори в нотації IDEF0 позначають інформаційні керуючі потоки. Слід підкреслити, що порядок відображення стрілок повинен чітко дотримуватися при формуванні моделей. Кожна сторона чотирикутника визначає тип стрілки. Порушувати ці правила не можна. Інакше створювані моделі не тільки не відповідають стандарту, але їх неможливо буде читати.

Усі стрілки починаються від краю діаграми і підходять до функцій. Таким чином, край діаграми в IDEF0 має змістове навантаження.

Отже, на рис. 4.13 показано основні принципи побудови діаграми в IDEF0.

На перший погляд, усе дуже просто. Проте з моменту появи нотації (у вигляді методології SADT) на початку 70-х рр. XX ст. вдаліших способів опису процесів організації на верхньому рівні не було запропоновано. Річ у тому, що найважливішою особливістю IDEF0 є можливість відображення

дій з управління, іншими словами, можливість опису управління процесами організації. Відмітимо, що відповідно до вимог стандарту IDEF0 для кожної функції на діаграмі повинно бути показано хоча б одна дія з управління. Це означає, що жодна функція без управління виконуватися не може.

Моделювання процесів у нотації IDEF0 починається зі створення так званої контекстної діаграми, яка описує діяльність організації або процесу в цілому. На контекстній діаграмі відображаються найважливіші входи і виходи, механізми, необхідні для роботи, а також керуючі дії.

Для розуміння принципів моделювання в IDEF0 розглянемо приклад побудови простої діаграми процесу.

Почнемо опис процесу з того, що помістимо на діаграму три функції, як показано на рис. 4.14.

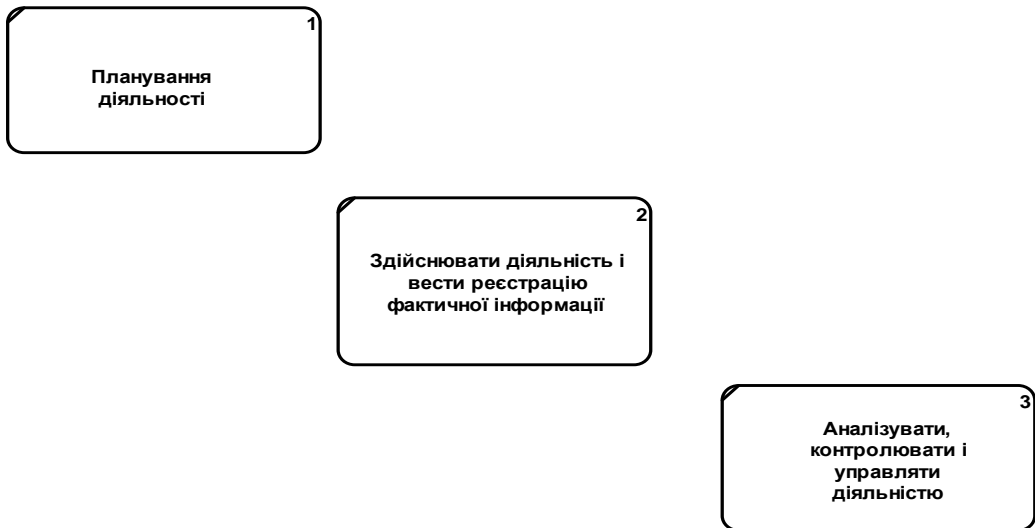


Рис. 4.14. Формування моделі бізнес-процесу. Крок 2

Першу функцію назвемо «Планувати діяльність», другу – «Здійснювати діяльність і вести реєстрацію фактичної інформації», третю – «Аналізувати, контролювати і управляти діяльністю». Звернемо увагу, що для назв функцій можуть бути використані тільки дієслова або віддієслівні іменники. Це одна з базових вимог нотації. Було б, наприклад, неправильно називати об'єкт «Начальник комерційного відділу» або «Відділ закупівель».

Найважливішими вимогами нотації є кількість об'єктів на діаграмі і кількість стрілок, що входять у кожную сторону чотирикутника. У стандарті рекомендовано розташовувати на одній діаграмі не більше шести і не менше двох функцій. З кожного боку в чотирикутник може входити не більше шести стрілок одночасно. Обидві ці вимоги обмежують кількість об'єктів на діаграмі процесу і примушують аналітика ретельніше продумувати схему створюваного процесу.

Об'єкти на діаграмі розташовані в шаховому, або так званому порядку домінування. Важливо відзначити, що цей порядок є зручним, і не слід від нього відступати. Необхідно підкреслити, що розташування об'єктів на діаграмі може не відповідати реальній послідовності виконання функцій. Річ у тому, що моделі IDEF0 призначені саме для опису процесів із погляду управління, а будь-які процеси управління системами є циклічними. Розглянемо рис. 4.15. Уявимо собі, що функцію планування виконує Комерційний відділ (КВ), який використовує при цьому засіб автоматизації MS Excel. Для планування КВ використовує інформацію про ринок (прайс-листи і т. п.) і заявки клієнтів. Регламентується діяльність КВ «Регламентом планування», «Планом організації на рік». Результатом роботи КВ є «План відвантаження ГП (готової продукції)». Подивимося, як ця інформація буде відображена на діаграмі.

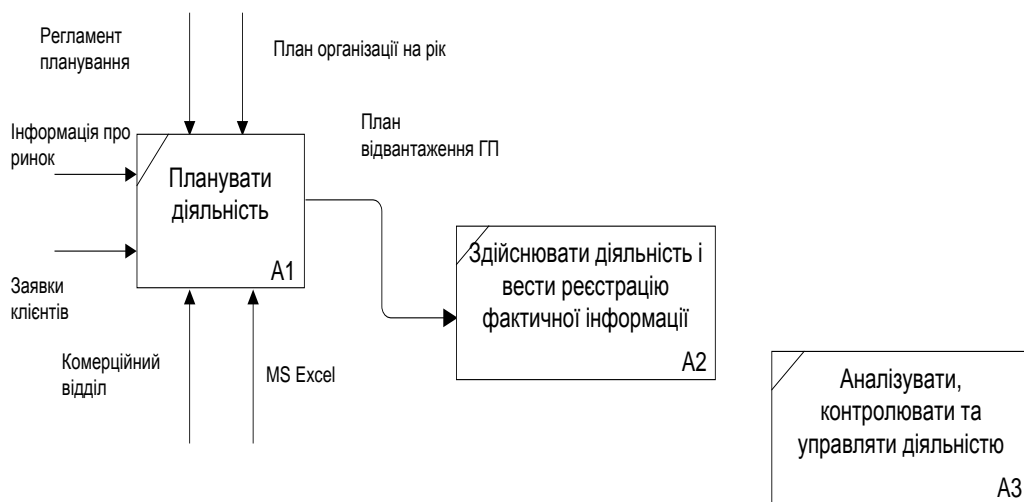


Рис. 4.15. Формування моделі бізнес-процесу. Крок 3

Розглянемо функцію «Здійснювати діяльність...». Її виконує Виробничий відділ (ВВ) і Цех. Для виконання робіт потрібна сировина і матеріали. Роботи регламентуються нормативами на витрату сировини державними, галузевими стандартами, технічними умовами, вимогами клієнта. Для роботи ВВ потрібна АСУТП власної розробки. Для виробництва готової продукції Цеху необхідні верстати та інше устаткування, тобто основні засоби (ОЗ).

Результат роботи ВВ і Цеху – готова продукція, яка є виходом функції «Здійснювати діяльність і вести реєстрацію фактичної інформації». Крім того, виходом цієї функції є також фактична інформація про виконання плану виробництва і відвантаження. На рис. 4.16 показано всі перераховані вище ресурси та інформація.

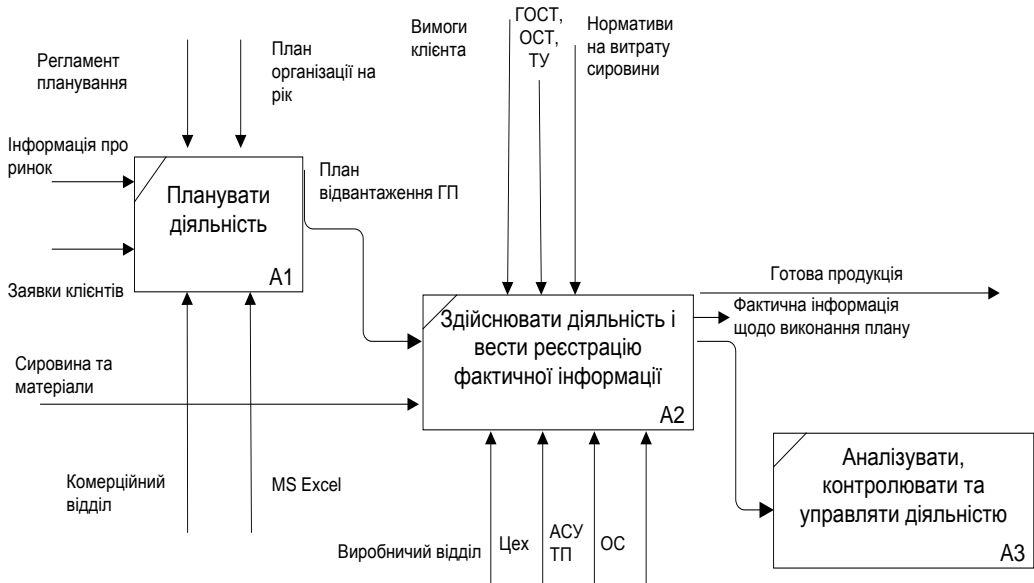


Рис. 4.16. Формування моделі бізнес-процесу. Крок 4

Нам залишилося показати входи і виходи функції «Аналізувати, контролювати і управляти діяльністю». Хто повинен її виконувати? Для цього прикладу вважатимемо, що контролює роботу той, хто її планує, тобто КВ.

У своїй роботі щодо аналізу і контролю КВ керується регламентом аналізу і контролю. Не варто забувати і річний план роботи організації в цілому. Для роботи КВ використовує MS Excel.

Судячи зі схеми процесу, представленої на рис. 4.16, КВ використовує вхід «Фактична інформація по виконанню плану». Що ще необхідне для виконання роботи КВ щодо аналізу і контролю? Звичайно, планова інформація, інакше ні з чим буде порівнювати фактичні дані і ухвалювати рішення. Таким чином, необхідно показати на схемі, що «План відвантаження ГП», який є виходом першої функції процесу і потрапляє на вхід функції «Здійснювати діяльність...», повинен також потрапляти і на вхід функції «Аналізувати, контролювати і управляти діяльністю». При цьому, як видно на рис. 4.17, стрілка, що зображує «План відвантаження ГП», розгалужується.

Результатом роботи КВ є звіт для керівництва організації «План/факт», як показано на рис. 4.17.

Як бачимо, стрілка, яка зображує КВ (як і MS Excel), не повторюється на діаграмі двічі. Вона розгалужується. Розгалуження стрілок – вдалий інструмент, що дозволяє зробити діаграму процесу більш наочною.

Отже, діаграма готова. Яким чином здійснюється управління цим циклічним процесом? Очевидно, що необхідно відобразити на схемі процесу принаймні два типи зворотних зв'язків за інформацією і управлінням (див. рис. 4.18).



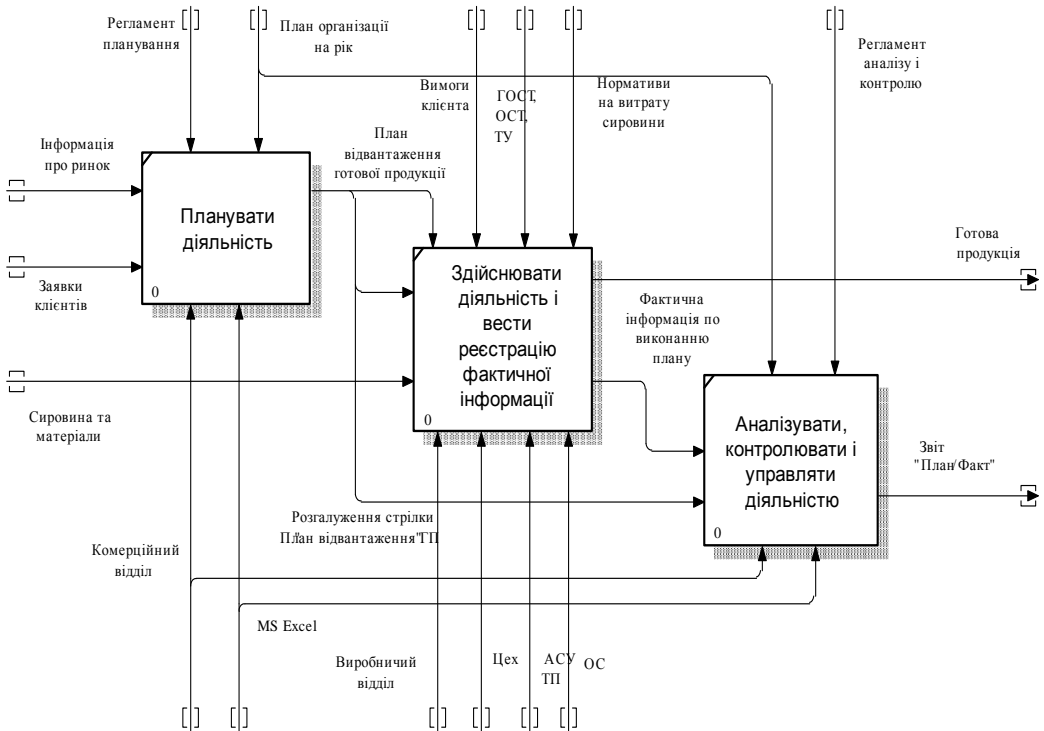


Рис. 4.17. Формування моделі бізнес-процесу. Крок 5

Першим типом зворотного зв'язку в діаграмах IDEF0 є зворотні зв'язки за інформацією. Вони зображені у вигляді стрілок, що виходять з правої сторони чотирикутника і входять в ліву сторону іншого чотирикутника. Зворотні зв'язки цього типу на діаграмі процесу обов'язково відображаються знизу, тобто обходять функції знизу.

У нашому прикладі покажемо зворотний зв'язок за інформацією «Інформація для коректування плану». Стрілка, що відображає цей зворотний зв'язок, виходить з правої сторони чотирикутника «Аналізувати, контролювати і управляти діяльністю» і входить в ліву сторону чотирикутника «Планувати діяльність». Таким чином, ми відобразили на діаграмі процесу той факт, що КВ регулярно аналізує виконання плану і, в разі відхилень від нього, формує інформацію, необхідну для коректування плану на наступний період.

Отже, зворотні зв'язки за інформацією дозволяють відобразити на діаграмах інформаційні потоки, необхідні для коректування дій, що виконуються по ходу бізнес-процесу.

Другим видом зворотного зв'язку є зворотний зв'язок з управління. Можливість відображення цих зворотних зв'язків – найважливіша перевага нотації IDEF0. Зворотний зв'язок з управління відрізняється від зво-

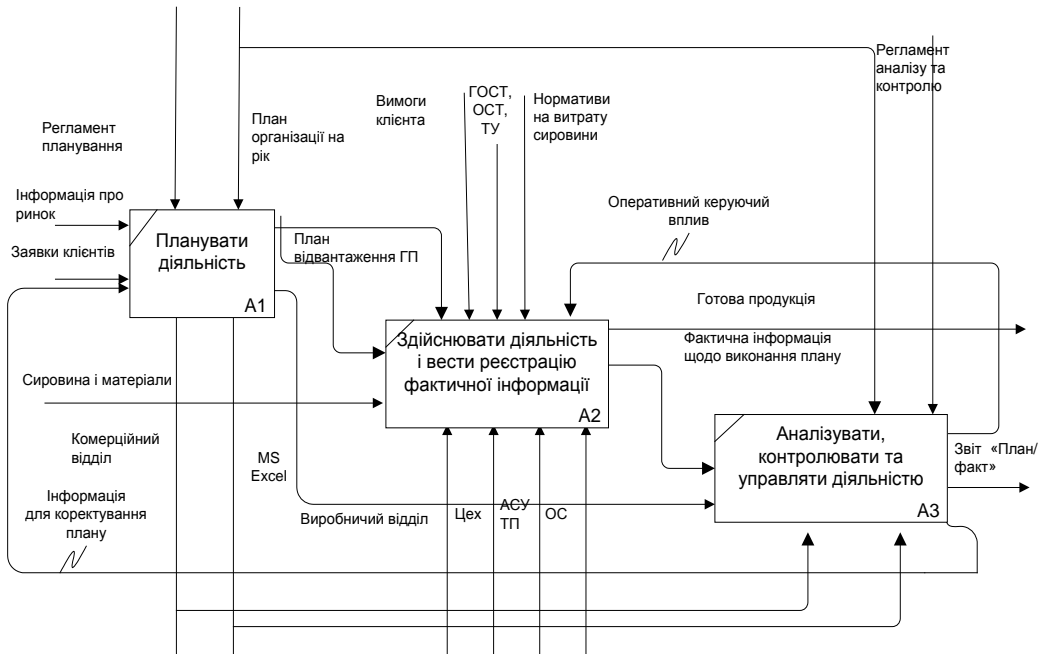


Рис. 4.18. Формування моделі бізнес-процесу. Крок 6

ротного зв'язку за інформацією тим, що стрілка, що зображає його на діаграмі, обходить її згори функцій і входить у верхню сторону чотирикутника.

У цьому прикладі покажемо зворотний зв'язок з управління «Оперативний керуючий вплив» у вигляді стрілки, що виходить з правої сторони чотирикутника «Аналізувати, контролювати і управляти діяльністю» і входить у верхню сторону чотирикутника «Здійснювати діяльність...». Цей зворотний зв'язок означає, що при аналізі і контролі виконання плану КВ ухвалює оперативні управлінські рішення, регулюючи виконання робіт ВВ і Цеху з виробництва готової продукції.

На рис. 4.18 ми додали ще одне розгалуження стрілки «План відвантаження ГП». Річ у тому, що ця стрілка може бути одночасно і інформаційним входом і входом з управління.

Розглянутий приклад показує, що при формуванні моделей процесів у IDEF0 можна (і потрібно!) ефективно використовувати стрілки, що відображають зворотні зв'язки за інформацією у управлінням.

Крім розгалуження, стрілки можуть також зливатися. Детально правила розгалуження і злиття стрілок описано в стандарті IDEF0. Тут ми наведемо декілька важливих прикладів використання цих правил.

На рис. 4.19 показано ситуації правильного і неправильного найменування стрілок при розгалуженні і злитті.

Розгалуження стрілок в ситуації 1 означає, що потік ресурсів А містить

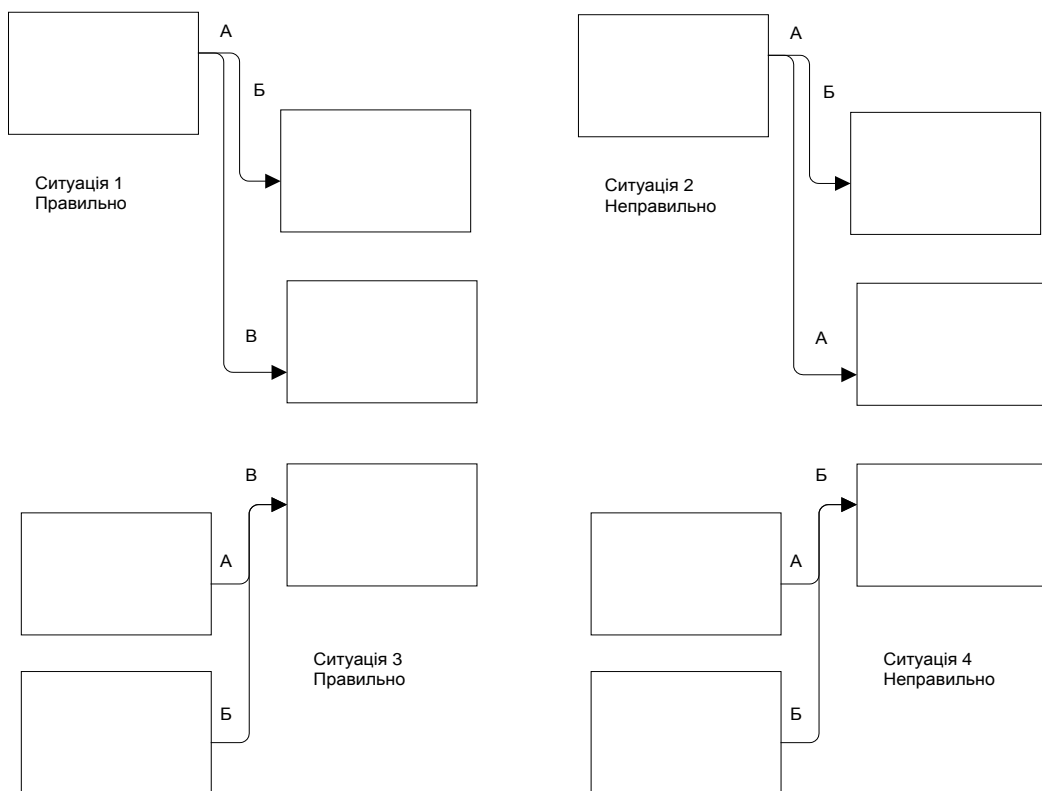


Рис. 4.19. Правила розгалуження і злиття стрілок

в собі потоки Б і В. Наприклад, план продажів може включати план із відвантаження в натуральному виразі і план із відвантаження у вартісному виразі.

Розгалуження стрілок в ситуації 2 неприпустимо, оскільки воно означало б, що потік А містить в собі одночасно і А і Б, що некоректно. Аналогічно можна розглянути ситуації 3 і 4 злиття стрілок.

На рис. 4.20 показано, як можна користуватися механізмом розгалуження і злиття стрілок при побудові діаграм процесів в IDEF0. Стрілка, що входить на діаграму процесу, розгалужується на кілька інших, які відображають детальніше потік ресурсів або інформації. Стрілки, що виходять, зливаються, показуючи, як формується результат виконання процесу в цілому. Сказане справедливо також для стрілок згори – керуючих дій і стрілок знизу – механізмів (персонал, інфраструктура).

Таким чином, розгалуження і злиття стрілок дозволяє показувати потоки ресурсів та інформації спочатку укрупнено, що важливе для опису процесів на верхньому рівні, а потім детальніше – для діаграм процесів нижнього рівня. Вказаний механізм ефективно використовується при побудові діаграм IDEF0 при декомпозиції моделей бізнес-процесів.

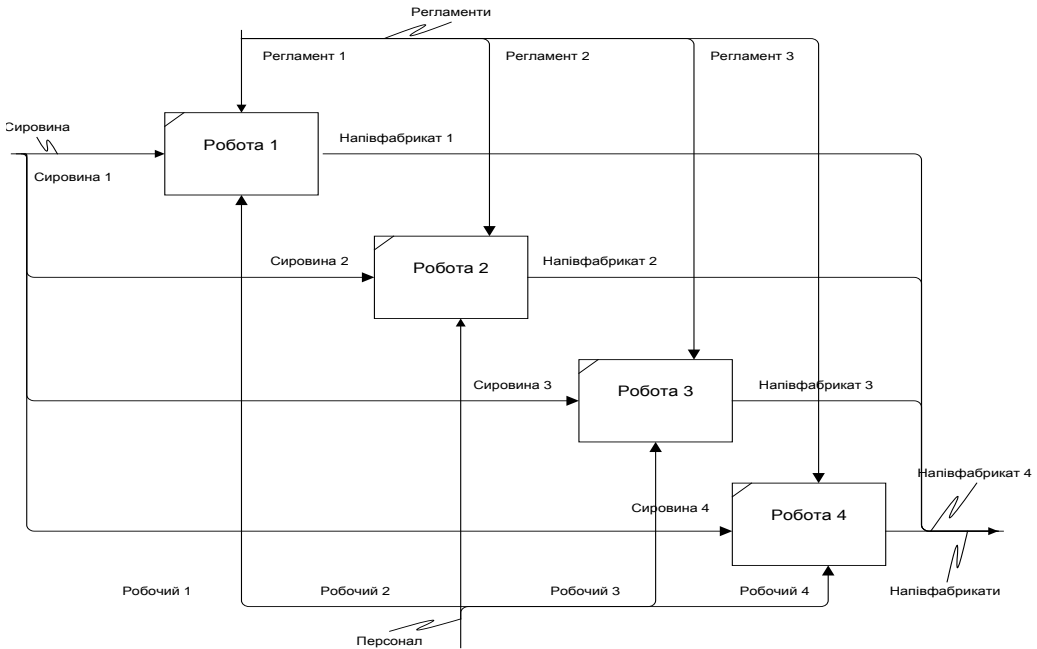


Рис. 4.20. Приклад розгалуження і злиття стрілок

Розгалуження і злиття стрілок є найважливішим інструментом для створення моделей в IDEF0. Особливо наочним цей факт стає при здійсненні декомпозиції моделей процесів з верхнього рівня на нижній.

Найважливішим поняттям нотації IDEF0 є поняття «тунелювання» стрілок. Виконаємо декомпозицію функції «Здійснювати діяльність...» (рис. 4.21). На детальнішому рівні ця функція включає наступні функції (роботи):

- «Розробляти графік виробництва»;
- «Виконувати підготовку виробництва»;
- «Виготовляти готову продукцію»;
- «Зберігати готову продукцію на складі»;
- «Відвантажувати готову продукцію клієнтам».

На першому кроці декомпозиції ми отримаємо схему процесу, на якій показано стрілки, що не входять ні жоден чотирикутник. Стрілки «мігрували» на рівень униз. Тепер необхідно «підв'язати» їх до конкретних функцій, при цьому можна використовувати механізм розгалуження і злиття стрілок. Звернемо увагу, що всі стрілки, показані на верхньому рівні, будуть показані і на нижньому рівні. Таким чином, зберігається зв'язність при моделюванні бізнес-процесу – детальні процеси виявляються однозначно пов'язаними з процесами верхнього рівня, і навпаки.

Тепер необхідно підвести кожен з показаних на рис. 4.21 стрілок до відповідного об'єкта – функції.

## Структурний аналіз бізнес-процесів

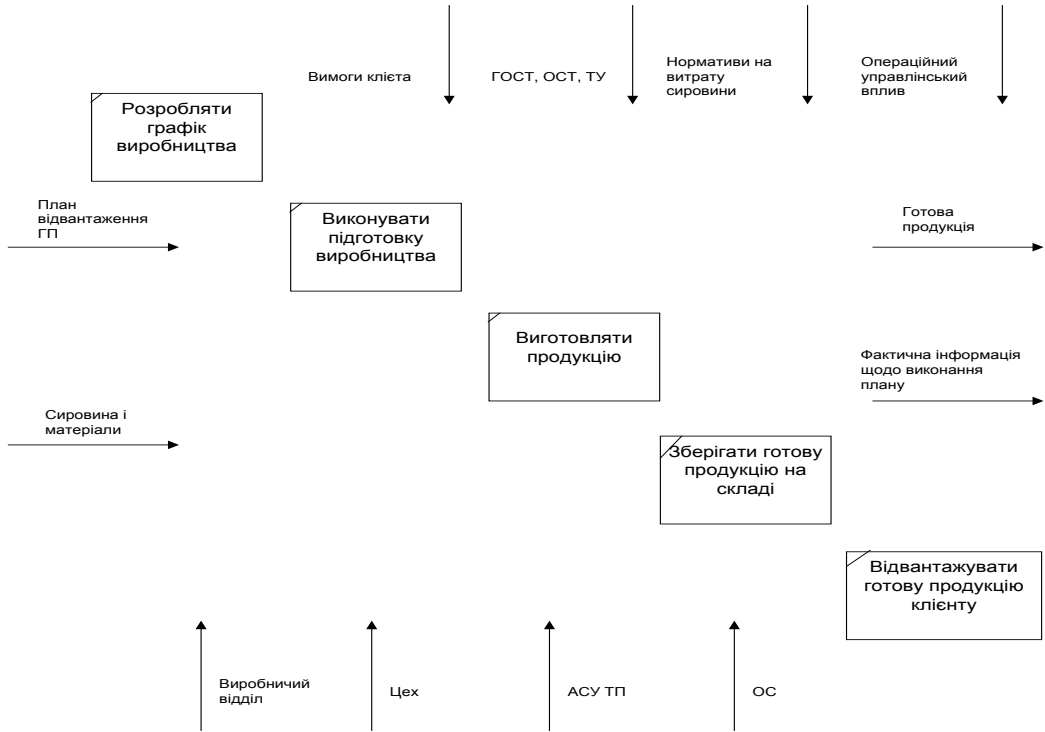


Рис. 4.21. Формування моделі бізнес-процесу. Крок 7

Функцію «План відвантаження ГП» підводимо до функції «Розробляти графік виробництва». До цієї функції згори також будуть підведені «Вимоги клієнта» і «План відвантаження ГП», але вже у вигляді дії, що управляє. Виходом першої функції є дія, що управляє, «Графік виробництва» та інформаційний потік «Дані графіка виробництва».

Вхідна стрілка «Сировина і матеріали» розгалужується на дві стрілки: «Допоміжна сировина» і «Основна сировина і матеріали».

Виходом другої функції процесу «Виконувати підготовку виробництва» є «Дані щодо готовності устаткування».

Третя функція процесу «Виготовляти готову продукцію» використовує вхідні матеріальні ресурси «Основна сировину і матеріали» та інформацію «Дані графіка виробництва» і «Дані щодо готовності устаткування». Виходами третьої функції є «Дані щодо виробництву ГП», «ГП на складі» (готова продукція, що відвантажується на склад) і «Брак». Звернемо увагу, що вихід «Брак» (стрілка і найменування виділені жирним шрифтом) не був показаний на діаграмі верхнього рівня, а з'явився тільки зараз, при докладному описі (рис. 4.22).

Постає питання, чому це могло відбутися. Займаючись описом процесу на верхньому рівні, ми цілком могли забути якийсь із виходів, або, вважаю-

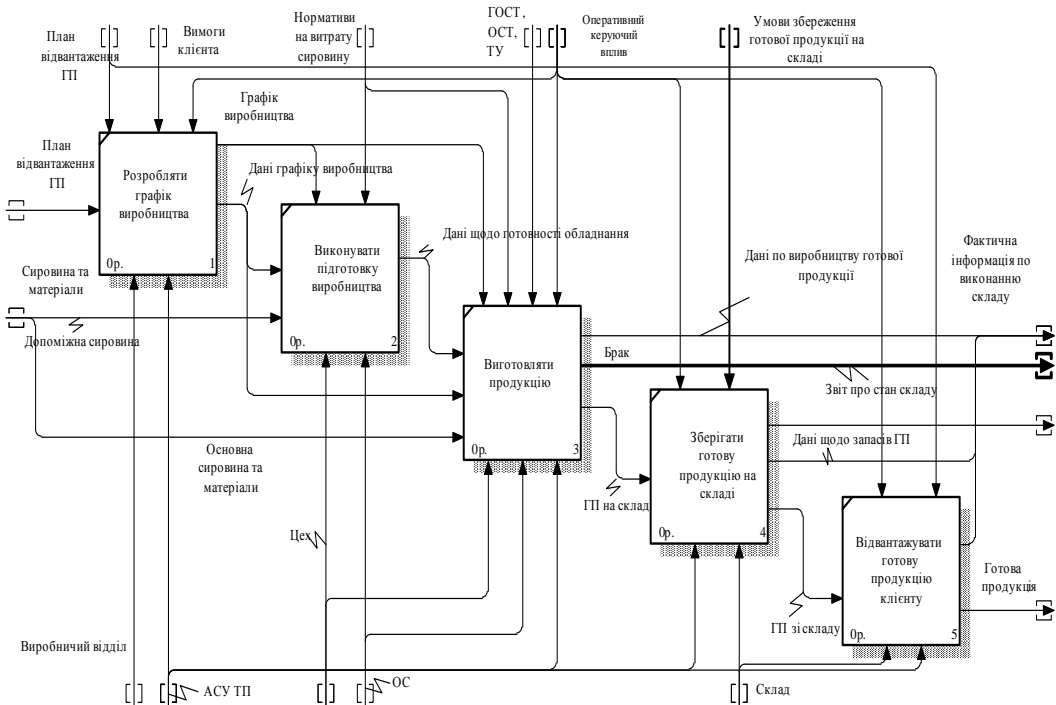


Рис. 4.22. Формування моделі бізнес-процесу. Крок 8

чи його малозначимим, проігнорували. На діаграмі процесу нижчого рівня цей вихід повинен бути відображений.

Четверта функція процесу «Зберігати готову продукцію на складі» формує виходи: «Дані щодо запасів ГП» і «ГП на складі». При її описі, проте, довелося додатково подати до розгляду і відобразити у вигляді стрілок виконавця «Склад ГП» і керуючий вхід «Умови зберігання ГП на складі».

Усі чотири нові входи, які були відсутні на діаграмі верхнього рівня і з'явилися на рис. 4.22, показано жирним шрифтом. Початок стрілки «Умови зберігання ГП на складі» поміщений у квадратні дужки. Це умовне позначення з'являється, коли ми показуємо на діаграмі нову стрілку нижчого рівня, якої немає на діаграмі верхнього рівня. Для стрілок, що входять в діаграму процесу, квадратні дужки показують на початку стрілки. Для нових стрілок, що з'являються, квадратні дужки указують в кінці, як, наприклад, для стрілки «Звіт за станом складу».

Квадратні дужки означають, що порушена нотація опису процесу. Щоб усунути виниклу суперечність з нотацією, необхідно або зробити стрілку «тунельною», або дозволити її міграцію на діаграму верхнього рівня. Так, наприклад, стрілка «Брак» зроблена «тунельною». Її не відображають на діаграмі верхнього рівня, вона буде видна тільки на поточній діаграмі. «Тунельні» стрілки беруть в круглі дужки.

У випадку із стрілкою «Склад ГП» — інша ситуація: ми вирішили протиріччя з нотацією, усунувши квадратні дужки і забезпечивши міграцію стрілки «Склад ГП» на діаграму верхнього рівня.

Таким чином, механізм тунелювання стрілок може бути ефективно використаний при проведенні декомпозиції бізнес-процесів. На діаграмах процесу верхнього рівня ми відображаємо потоки ресурсів та інформації укрупнено. При декомпозиції щоразу ми можемо відображати все більш детальні потоки, при цьому схема процесу ускладнюється за рахунок більшої кількості стрілок.

Слід зазначити, що тунелювання стрілок зазвичай використовують одночасно з розгалуженням, що забезпечує зв'язність і прозорість діаграм процесів без зайвого ускладнення.

Кожен об'єкт (функція, робота) на діаграмі процесу в нотації IDEF0 може бути пронумерований. Існує декілька способів нумерації. Ми розглянемо найбільш простий і часто вживаний спосіб – дерево функцій процесу (рис. 4.23), на прикладі процесу, представленого вище.

Як видно з рис. 4.23, нумерація діаграм йде згори вниз – від діаграми верхнього рівня до діаграм нижнього рівня. Кожна діаграма нижнього рівня отримує свій номер на основі номера батьківської діаграми верхнього рівня. Наприклад, функції «Здійснювати діяльність...» присвоєно номер А2, а функції процесу нижчого рівня – номери А21–А25. Якщо ми декомпуємо функцію А22, то функції детальнішого процесу буде присвоєно номери А221–А22М. Буквений індекс А вводиться умовно. Використання

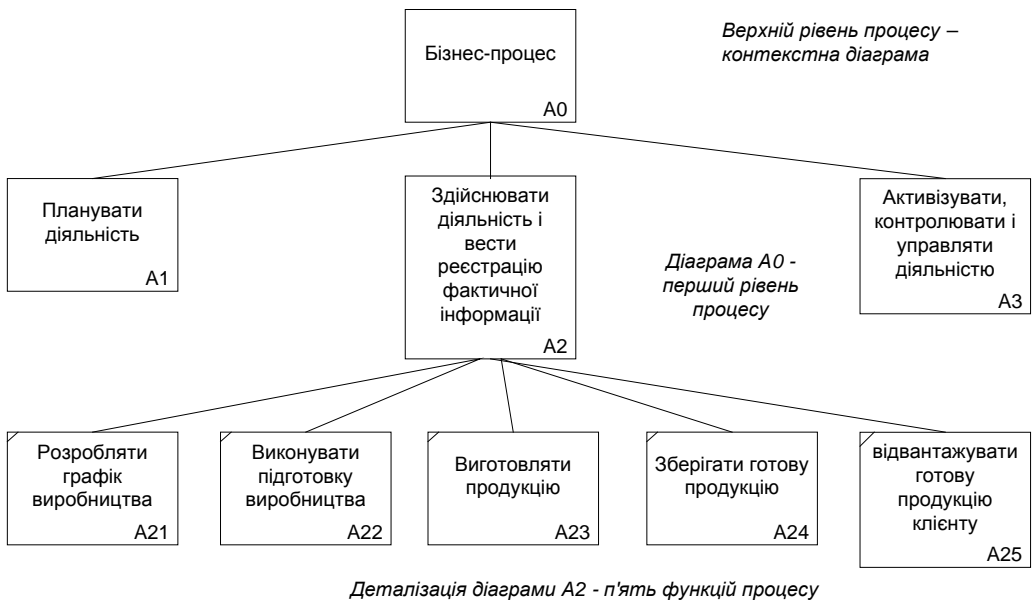


Рис. 4.23. Діаграма дерева функцій

розглянутого механізму нумерації робить відстежування функцій процесів досить наочним. Нагадаємо, що кількість функцій на одній діаграмі не повинна складати більше шести (іноді допускається вісім). В цьому випадку за номером функції завжди можна однозначно визначити рівень процесу.

На рис. 4.24 представлено діаграму процесу, взяту в так звану рамку IDEF0. Вгорі і знизу діаграми розташовані декілька полів для відображення інформації про діаграму процесу. Розглянемо спочатку верхні поля діаграми.

Поле USED AT використовують для вказівки посилань на інші місця моделі (інші діаграми), в яких є посилання на цю діаграму.

Вказівки про автора діаграми, найменування проекту, в ході реалізації якого була створена діаграма, дати створення і останнього перегляду просявляють на полях Author, Project, Date, Rev.

Поле Notes використовують при перевірці моделі експертом. Порядок роботи в цьому випадку наступний. Автор діаграми передає її експертові, із слів якого було побудовано опис процесу. Експерт читає діаграму і в разі незгоди із схемою процесу робить свої зауваження письмово, безпосередньо на діаграмі. Кожне зауваження повинно бути пронумероване. При вказівці зауваження експерт обводить порядковий номер зауваження в полі Notes. Такий порядок розроблений для того, щоб автор моделі – аналітик – міг усунути всі зауваження, чітко контролюючи їх кількість. Кількість виправлень повинна відповідати кількості зауважень.

Далі йдуть поля статусу діаграми: Working, Draft і т. д. На кожному полі вказують дату, а також просявляють підпис особи, уповноваженої змінювати статус діаграми. Діаграми, що знаходяться в роботі, отримують статус Working. Діаграми, що затверджені і є обов'язковими для виконання, можуть отримати, наприклад, статус Publication.

У полі Context вказують номер діаграми верхнього рівня, що містить процес, який розглядається на цій діаграмі, у вигляді однієї з функцій. Крім того, на цьому полі графічно показано положення цієї діаграми серед функцій діаграми верхнього рівня.

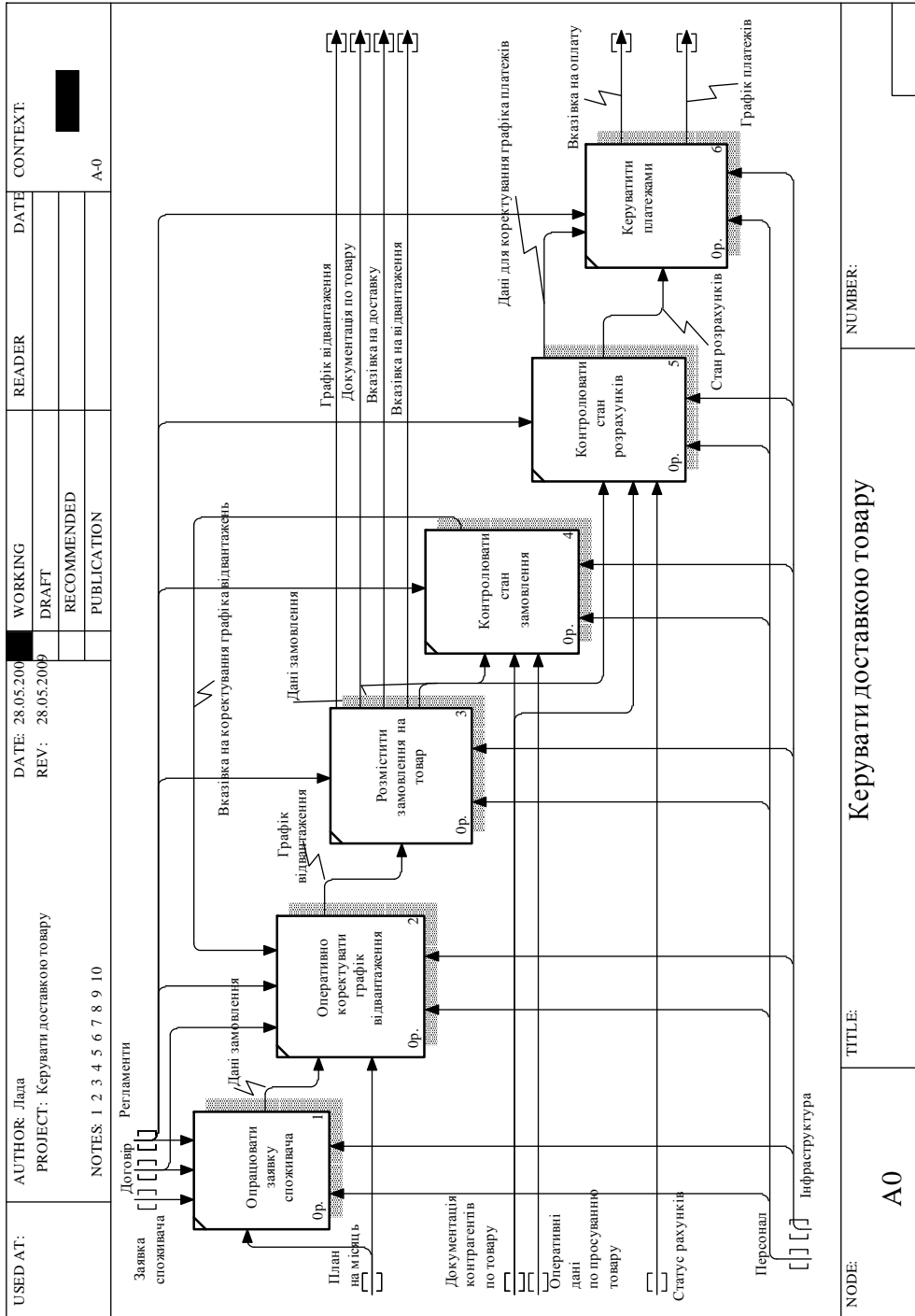
Розглянемо поля, що знаходяться в нижній частині рамки діаграми IDEF0.

На полі Node просявляють номер вузла, присвоєний цій діаграмі (нумерація діаграм розглянута вище).

Потім слідує поле Title, де вказують назву діаграми. Відмітимо, що назва діаграми збігається з назвою декомповованої функції діаграми верхнього рівня.

На полі Number просявляють унікальний номер діаграми, який привласнює автор. На полі без назви вказують номер листа з діаграмою в підшивці документів (для формування звіту, що містить декілька діаграм).





Керувати доставкою товару

Рис. 4.24. Рамка IDEFO

NUMBER:

TITLE:

A0

Рамка IDEF0 – зручний стандартний інструмент для вказівки основних характеристик діаграми бізнес-процесу. Дані, що наводяться в ній, однозначно визначають положення діаграми стосовно інших діаграм, поточний статус, дату останнього перегляду тощо.

Підкреслимо, що наявність стандартної рамки робить методологію IDEF0 ще зручнішим інструментом для опису бізнес-процесів, оскільки в усіх сучасних системах моделювання процесів (наприклад, BPWin 6), що підтримують IDEF0, більшість найважливіших полів рамки заповнюються автоматично. Це робить процес документування моделей досить простим і прозорим та істотно полегшує роботу аналітиків при створенні комплекту моделей бізнес-процесів організації [15,16].

#### **4.2.3. Переваги і недоліки використання IDEF0 для опису бізнес-процесів**

Методологія моделювання бізнес-процесів IDEF0, на наш погляд, призначена для опису процесів верхнього рівня. Описуючи такі процеси, аналітик приділяє величезну увагу управлінню процесами, зворотним зв'язкам з управління та інформації. Наведемо основні переваги і недоліки методології IDEF0:

##### **ПЕРЕВАГИ**

- Повнота опису бізнес-процесу (управління, інформаційні та матеріальні потоки, зворотні зв'язки)
- Комплексність при декомпозиції (міграція і тунелювання стрілок)
- Можливість агрегування та деталізації потоків даних та інформації (розділення і злиття стрілок)
- Наявність жорстких вимог методології, що забезпечують отримання моделей процесів стандартного виду
- Простота документування процесів
- Відповідність підходів до опису процесів у IDEF0 MC ISO 9000:2000

##### **НЕДОЛІКИ**

- Складність сприйняття (велика кількість стрілок)
- Велика кількість рівнів декомпозиції
- Труднощі зв'язності декількох процесів, наданих у різних моделях однієї і тієї ж організації

Найважливішою характерною межею IDEF0 є повнота опису бізнес-процесу, яка досягається за рахунок наявності засобів, що відображають дії з управління зворотні зв'язки з управління та інформації. Методологія IDEF0 надає аналітику можливість не піклуватися про комплексність декомпозиції шляхом використання механізмів міграції і тунелювання стрілок. Такий механізм забезпечує зв'язність створюваних діаграм між собою. Крім того, він робить модель процесу наочною. Використання можливості розділення і злиття стрілок також сприяє створенню більш наочних та відпрацьованих моделей. Резюмуючи, можна сказати, що жорсткі вимоги щодо формуванню моделей у IDEF0 в поєднанні з гнучкими засобами пред-

ставлення потоків інформації та ресурсів забезпечують створення IDEF0-моделей стандартного вигляду (див. практикум).

Основною перевагою методології IDEF0 є також відповідність формату представлення процесу в IDEF0 визначенню процесу МС ІСО 9000:2000, що дозволяє вибирати IDEF0 в якості внутрішнього стандарту організації, який регламентує опис бізнес-процесів [39].

До недоліків IDEF0 можна віднести складність сприйняття схем процесів співробітниками організації, особливо керівниками. Слід зазначити, проте, що ефективне застосування будь-якої нотації припускає навчання, як співробітників, так і керівників уміння читати і аналізувати схеми процесів.

Крім того, застосовуючи IDEF0, складно пов'язувати між собою моделі декількох процесів (наприклад, збут і виробництво) за необхідності створення окремих моделей для кожного з цих процесів. Проте недолік є, скоріше за все, технічним і може бути усунений шляхом попередніх домовленостей про правила моделювання [32].

На практиці часто зустрічаються ситуації, коли моделі IDEF0 використовуються для опису послідовно виконуваних робіт. У таких моделях, як правило, слабо відображено управління процесом, не вказані керівники, майже немає зворотних зв'язків. На наш погляд, використовувати IDEF0 для опису послідовно виконуваних робіт некоректно.

### 4.3. Методика IDEF3

Нотація IDEF3 є другою найважливішою нотацією (після IDEF0) і призначена для опису потоків робіт (Work Flow Modeling). IDEF3 широко використовується для створення моделей бізнес-процесів організації на нижньому рівні – при описі робіт, що виконуються в підрозділах і на робочих місцях. Слід зазначити, що нотація IDEF3 була узята за основу при створенні методики опису процесів ARIS eEPC – «розширеного ланцюжка процесу, керованого подіями» [28,35].

Основними графічними об'єктами моделі, використовуваними в IDEF3, є чотирикутники і стрілки. Перші слугують для опису функцій (робіт, процесів) другі – для віддзеркалення в моделі послідовності виконання, функцій у часі або послідовності виконання функцій, обумовленої потоком матеріальних ресурсів. Перш ніж перейти безпосередньо до нотації IDEF3, розглянемо наступний приклад. На рис. 4.25 представлено два варіанти можливого опису потоку робіт.

На рис. 4.25 (варіант 1) показано, що спочатку виконується функція 1. Після завершення виконання цієї функції починають одночасно викону-

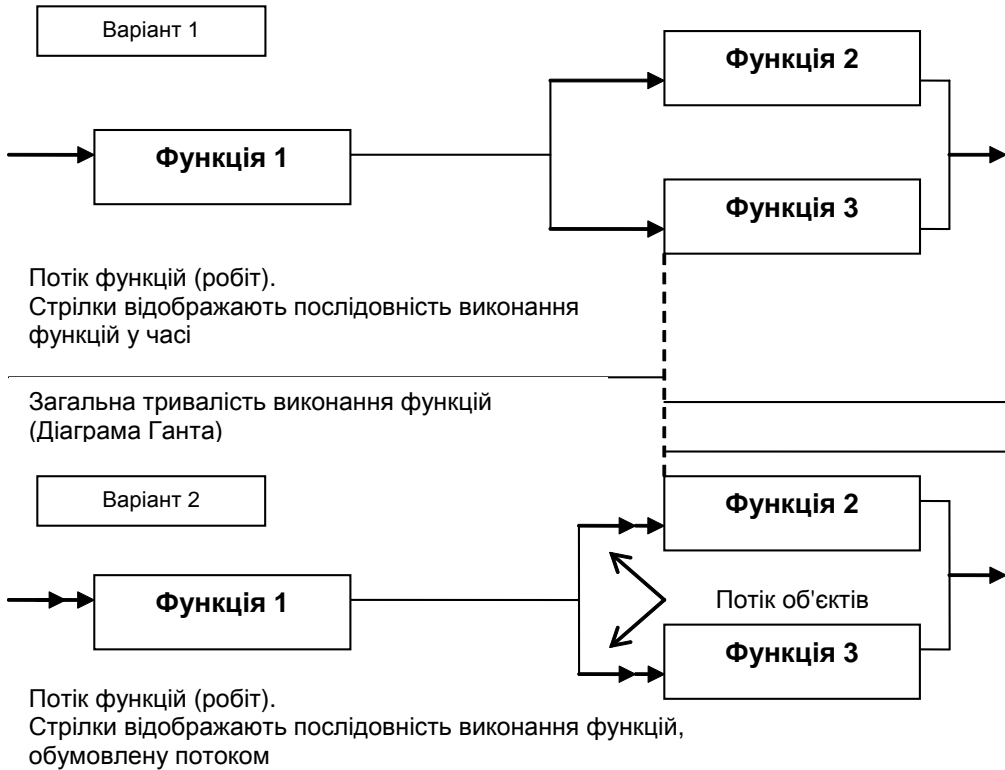


Рис. 4.25. Опис потоків робіт

ватися функції 2 і 3. Стрілки в цьому випадку показують нам, яким чином завершення виконання однієї функції впливає на початок виконання іншої.

Процес варіанта 2 побудований інакше. Початок виконання функцій тут обумовлений надходженням на вхід деяких матеріальних ресурсів (вхід функції 1), закінчення – виходом матеріальних ресурсів (вихід функції 1). Потоки ресурсів визначають початок виконання наступних функцій процесу (функцій 2 і 3) і т. д.

З'ясуємо, у чому полягають недоліки способів опису процесів, представлених на рис. 4.25. Річ у тому, що побудовані таким чином схеми процесів неможливо однозначно зрозуміти (прочитати). Функції 2 і 3 можуть виконуватися не одночасно. Наприклад, може скластися ситуація, коли буде потрібно виконання або функції 2, або функції 3 процесу. Очевидно, що в цьому випадку обраний нами спосіб опису процесу не дозволить зробити висновок, який же варіант розвитку подій реалізується насправді. Повернемося до нотації IDEF3.

Для того щоб уникнути неоднозначності опису потоків робіт, в нотації IDEF3 визначені додаткові об'єкти, які слугують для відображення мож-

ливих варіантів розгалуження і злиття потоків робіт, що реалізуються за певних умов. Вказані об'єкти є логічними символами трьох видів:

- логічний оператор «І»;
- логічний оператор «АБО»;
- логічний оператор, що виключає «АБО».

На рис. 4.26 показано приклад застосування логічного оператора «І». Процес починається з функції, після якої стоїть знак логічного оператора «І», тобто перехрестя. Після перехрестя процес розгалужується, і одночасно починають виконуватися наступні дві функції процесу. Після того як вони виконані, відбувається злиття стрілок процесу за допомогою значка «І». Це означає, що остання функція процесу починає виконуватися тоді, коли закінчено виконання двох попередніх функцій.

На рис. 4.27 представлено модель з логічним оператором «АБО». Такий оператор означає, що після виконання першої функції процесу можуть відбутися три події: 1) виконується функція 2; 2) виконується функція 3; 3) виконуються функції 2 і 3 одночасно.

Рис. 4.28 демонструє застосування логічного символу, що виключає «АБО». В такому випадку після виконання функції 1 може початися виконання або функції 2, або функції 3. Далі, після виконання якої-небудь з цих функцій, ми знову потрапляємо на перехрестя, тобто логічний оператор,

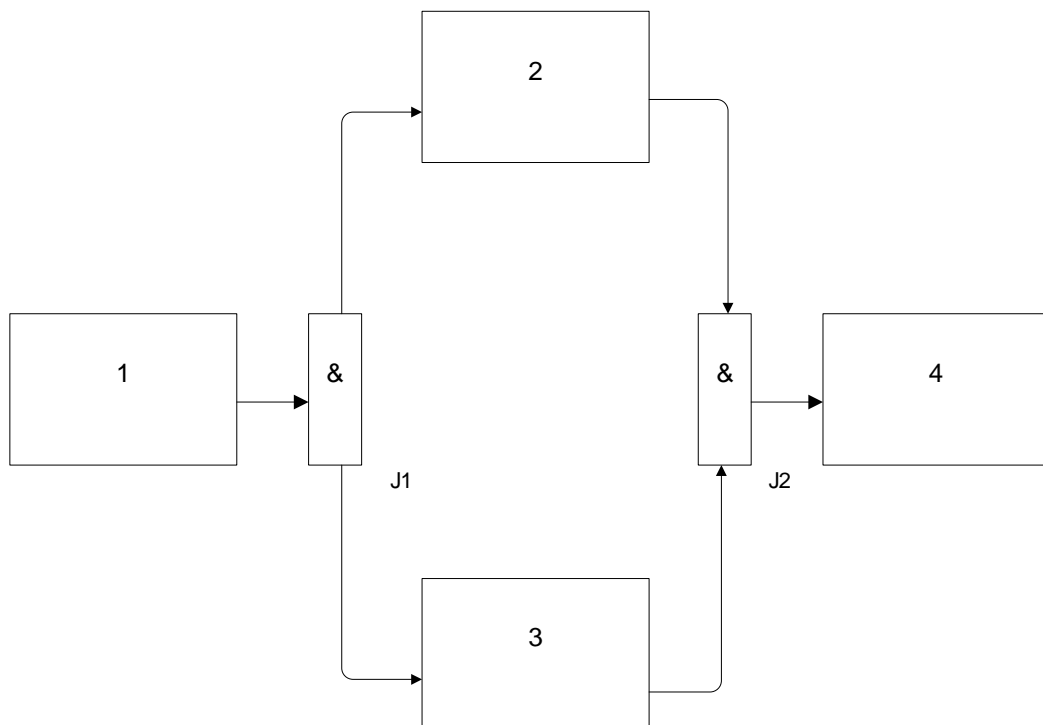


Рис. 4.26. Модель процесу з логічним оператором «І»

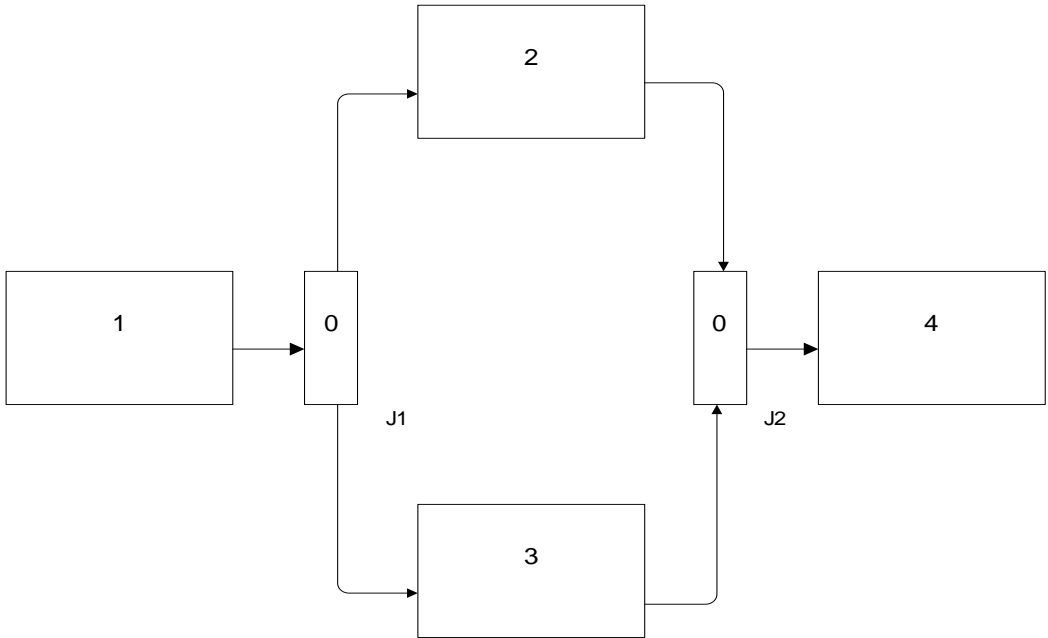


Рис. 4.27. Модель процесу з логічним оператором «АБО»

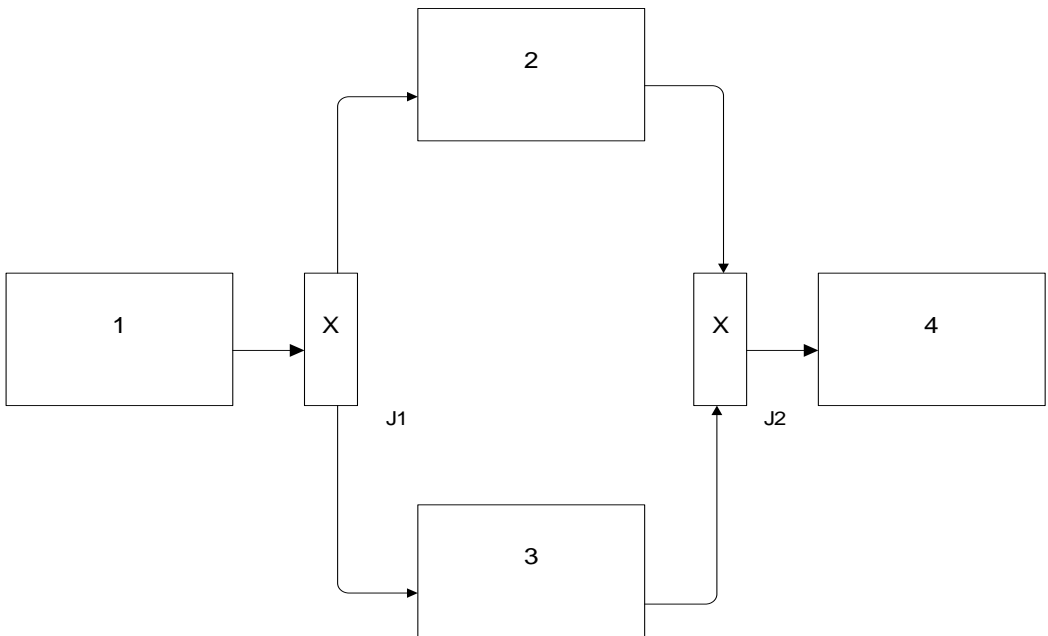


Рис. 4.28. Модель процесу з логічним оператором, що виключає «АБО»

Виконання двох функцій (2 і 3) повинно починатися одночасно після завершення виконання функції 1

Знак асинхронного «!» показує, що виконання функцій 2 й 3 може закінчитися не одночасно, при цьому виконання функції 4 не почнеться, поки не виконані функції 2 і 3

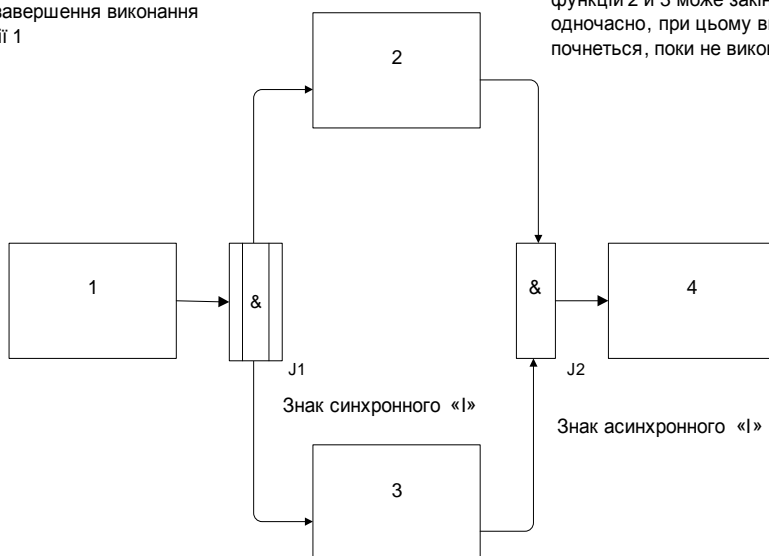


Рис. 4.29. Модель процесу з логічним оператором «!»

що виключає «АБО». Функція 4 буде виконана або після закінчення функції 2, або функції 3.

Логічні оператори можуть бути синхронними і асинхронними. На рис. 4.29 показана різниця між синхронним і асинхронним логічним оператором «!».

На відміну від нотації IDEF0, в нотації IDEF3 сторони чотирикутника, що зображає функцію (роботу, процес), не використовують для прив'язки входів різного типу. Більше того, в чотирикутник може входити і виходити тільки одна стрілка. Інакше правила побудови діаграм в IDEF3 будуть порушені.

При декомпозиції процесів в IDEF3 не відбувається міграції і тунелювання стрілок. Аналітик повинен сам піклуватися про зв'язність моделювання процесу і коректності декомпозиції. Можливий приклад декомпозиції функції «Виконувати підготовку виробництва» з нотації IDEF0 на процес в нотації IDEF3 показано на рис. 4.30. Зауважимо, що функція «Отримати допоміжну сировину на складі» ініціюється надходженням затвердженого графіка виробництва. Цей факт відображений вхідною стрілкою «Графік виробництва». На діаграмі процесу показана також стрілка «Допоміжна сировина». Подібне її уявлення є порушенням нотації опису. Проте загалом таким прийомом можна користуватися, не забуваючи при цьому змінювати тип стрілки на стрілку з двома наконечниками, що відображає потік об'єктів (матеріальних ресурсів або інформації).

На рис. 4.31 наведено приклад бізнес-процесу в нотації IDEF3 під назвою «Обробити заявку клієнта». Такий процес є частиною більш загаль-

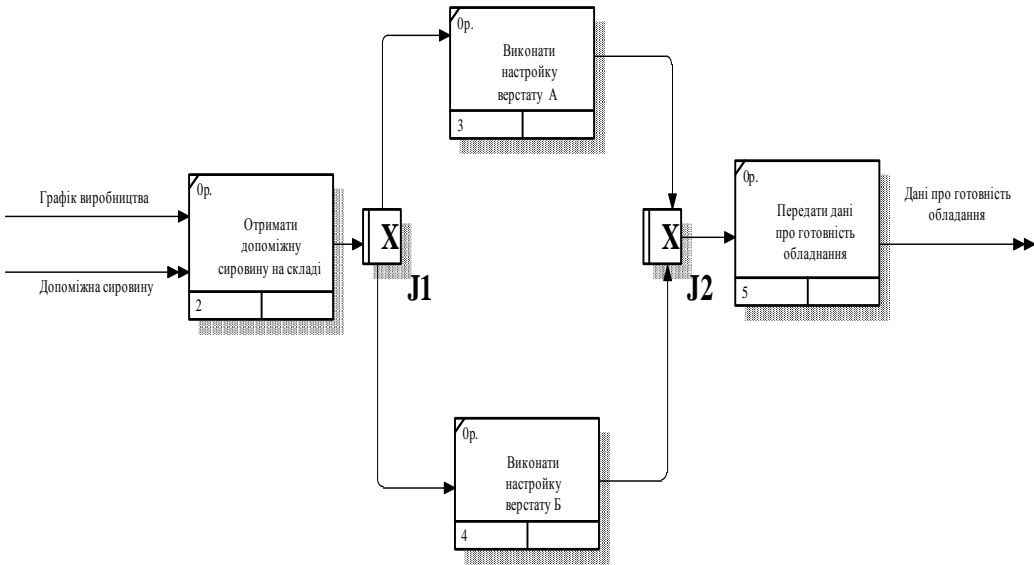


Рис. 4.30. Приклад моделі процесу в стандарті IDEF3

ного процесу «Збут готової продукції». Процес починається з надходження заявки клієнта на вхід функції «Виконати облік замовлення в системі». По ходу її виконання дані замовлення клієнта реєструються в системі автоматизації (наприклад, в MS Excel). Потім менеджер відділу збуту виконує перевірку на відповідність номенклатурі (функція «Виконати аналіз на відповідність номенклатурі»). Результатом виконання цієї функції можуть бути два моменти: перший – «замовлення відповідає номенклатурі», другий – «замовлення не відповідає номенклатурі». Для віддзеркалення цих подій в моделі процесу використовують логічний оператор, що виключає «АБО».

Після цього логічного оператора процес розгалужується. У разі невідповідності замовлення номенклатурі виконується нижня гілка процесу, а саме функції: «Повідомити клієнта про неможливість виконання замовлення» і «Внести замовлення клієнта до статистики незадоволеного попиту».

Якщо замовлення клієнта відповідає номенклатурі, починають рух по верхній гілці процесу. Виконується функція «Погоджувати заявку з ПЗВ». До цієї функції прив'язаний посилальний об'єкт «Погоджувати з ПЗВ у разі відповідності заявки номенклатурі». ПЗВ аналізує замовлення і робить висновок про можливість його реалізації. Наприклад, може скластися ситуація, коли не вистачає виробничих потужностей через ремонт, невідповідність величини замовлення економічно обґрунтованим розмірам партії тощо. В цьому випадку знову переходять на нижню гілку процесу, при цьому використовують логічний оператор «АБО». Він слугує для об'єднання можливих входів у функцію «Повідомити клієнта про неможливість виконання замовлення».

Якщо ПЗВ вважає замовлення здійсненим, то проводять детальний роз-



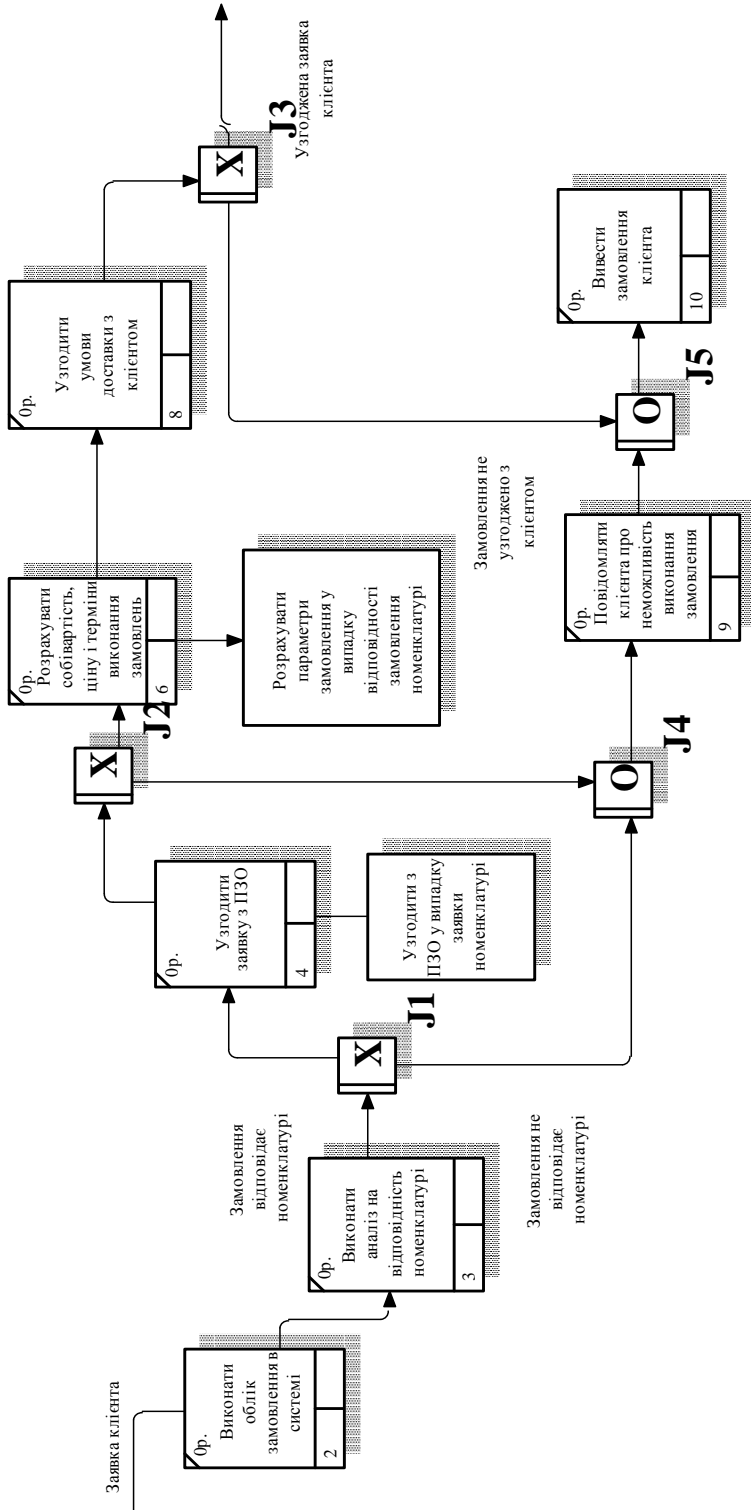


Рис. 4.31. Модель бізнес-процесу «Обробити заявку клієнта» у нотатції IDEF3

рахунок собівартості виконання і визначають його ціну. Встановлюють також терміни виконання замовлення (функція «Розрахувати собівартість, ціну і терміни виконання замовлення»). Далі розрахункові цифри погоджують з клієнтом – виконується функція «Погоджувати умови постачання з клієнтом». Знову можливі два варіанти – використовують логічний оператор, що виключає «АБО». У разі якщо клієнта не влаштовують фінансові умови, то він відмовляється від замовлення, а замовлення вносять до статистики незадоволеного попиту (нижня гілка процесу).

Якщо клієнт готовий працювати на запропонованих умовах, то процес закінчується. Виходом процесу слугує «Узгоджена заявка клієнта» і дані за розрахованими параметрами замовлення (на схемі процесу не показані).

Аналіз процесу, представленого на рис. 4.31, наводить на думку про те, що нотацію IDEF3 доцільно застосовувати вразі відносно простих процесів на нижньому рівні декомпозиції, тобто процесів рівня робочих місць. В цьому випадку схема процесу може стати основою для створення документів, що регламентують роботу виконавців. Очевидно, що процес у нотації IDEF3 є «пласким». За допомогою цієї нотації достатньо складно створювати комбіновані моделі, в яких би поєднувалися описи потоків робіт і процеси управління цими роботами. Цей факт стає очевидним особливо при порівнянні описів процесів у нотації IDEF3 і IDEF0 [15,16].

#### **4.4. Моделювання процесів у нотації DFD**

Одним із найважливіших способів опису процесу є діаграми потоків даних (інформації) DFD (Data Flow Diagram). Подібні діаграми містять, як правило, два типи графічних об'єктів: чотирикутники і стрілки. Перші описують функції (роботи, процеси), другі – потоки даних між цими функціями. Проста схема процесу у форматі DFD показана на рис. 4.32.

На діаграмі DFD функції зазвичай розташовуються зліва направо в порядку, відповідному послідовності їх виконання в часі, хоча це не є обов'язковим. Якщо дотримуватися вказаної вимоги, то отримана схема – це опис процесу, який схожий з описом процесу в нотації IDEF3. Таким чином, до опису процесів у DFD застосовні типові правила декомпозиції. Що стосується сторін чотирикутників, то в нотації DFD вони не мають того значення, як в IDEF0. Слід зазначити, що існує декілька підходів до формування моделей потоків даних. Тут ми розглядаємо нотацію DFD, реалізовану в інструментальному середовищі BPWin [15, 16].

Часто нотацію DFD плутають з простим описом потоків інформації між

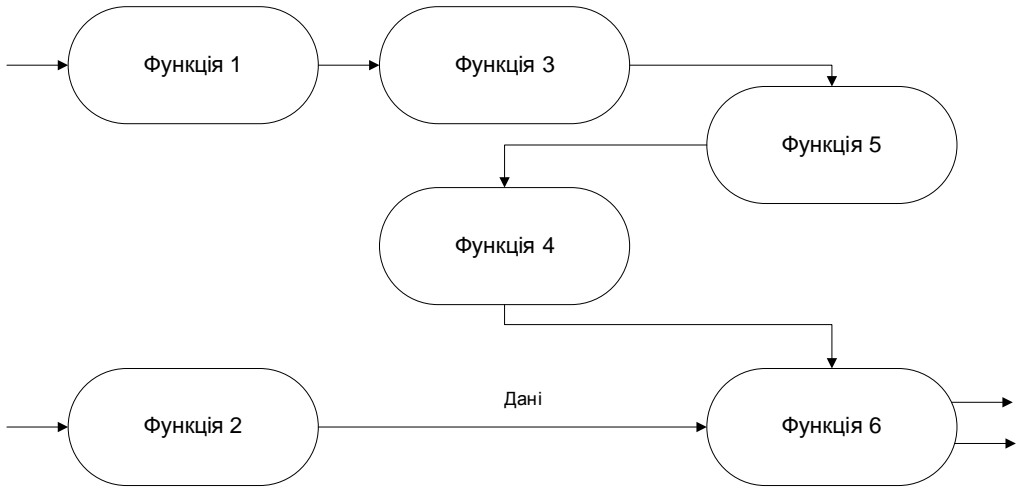


Рис. 4.32. Приклад найпростішої моделі потоків даних

підрозділами. Насправді, це не так. На рис. 4.33 представлено модель, яка відображає потоки даних між підрозділами, але не є моделлю процесу.

Виникає питання, чому не можна розглядати простий опис потоків між підрозділами організації як схему процесу? Відповідь достатньо проста. У кожному великому підрозділі (наприклад, відділ збуту великого підприємства) виконуються різні бізнес-процеси. Часто у цих процесів існують різні внутрішні і зовнішні клієнти. Саме тому схема, наведена на рис. 4.33, описує тільки потоки даних, що перетинають межі підрозділів, але не містить інформації про реально виконувані бізнес-процеси як на рівні підрозділів, так і на рівні організації в цілому. До речі, розглянутий на рис. 4.33 формат представлення потоків даних є практичним і широко використовуваним.

Приклад опису процесу в DFD можна ускладнити, використовуючи по-

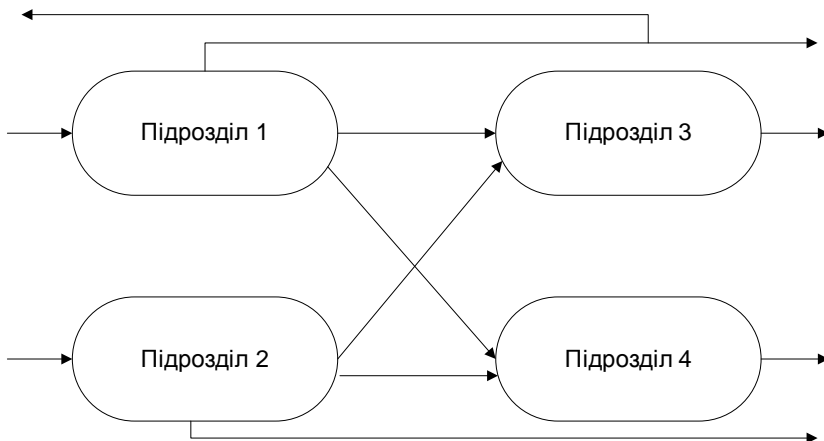


Рис. 4.33. Приклад моделі потоків даних між підрозділами організації

няття «Сховище даних». Під ним розуміється будь-який носій інформації, наприклад, паперовий документ, електронний файл, промислова база даних на сервері організації і т.д. При побудові моделі процесу з використанням сховищ даних необхідно пам'ятати, що дані (інформація) не можуть переміщатися між функціями процесу самі по собі. Їх можна передавати тільки через певних посередників – носіїв інформації або, що те ж саме, сховищ даних. На рис. 4.34 представлено модель процесу в нотації DFD, побудовану з використанням поняття «Сховище даних».

Навіщо потрібні нотації DFD? В першу чергу – для опису реально існуючих в організації потоків даних. Описи можуть створюватися як за процесною, так і за функціональною ознакою. У першому випадку ми отримуємо моделі бізнес-процесів у форматі DFD, в другому – схему обміну даними між підрозділами. Створені моделі потоків даних організації можуть бути використані при розв'язанні таких задач, як [44]:

- визначення існуючих сховищ даних (текстові документи, файли, система управління базою даних (СУБД));
- визначення і аналіз даних, необхідних для виконання кожної функції процесу;
- підготовка до створення моделі структури даних організації, так званої ERD-модель (IDEF1X);

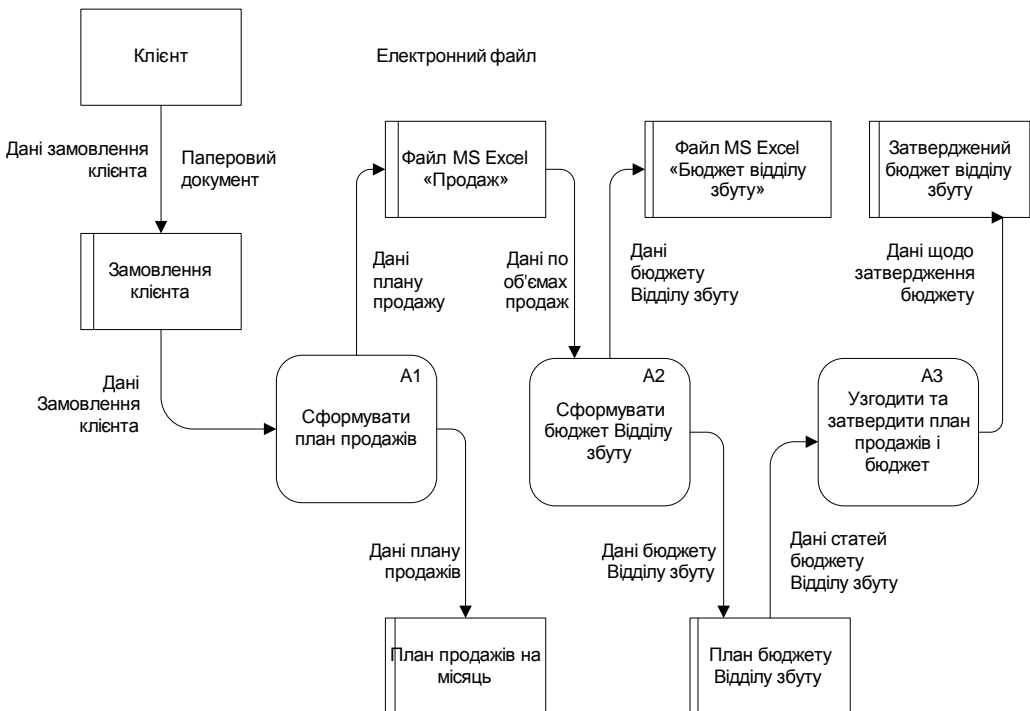
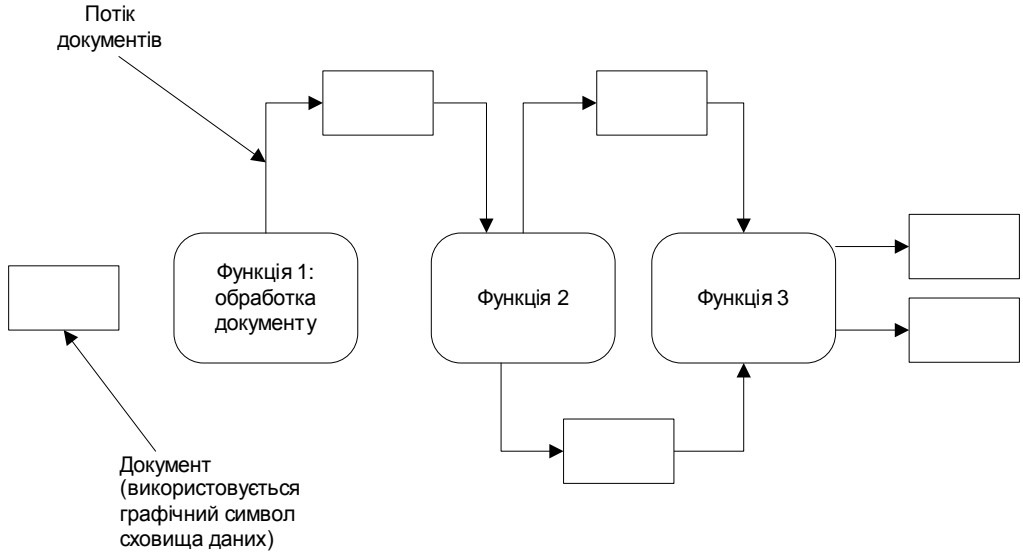
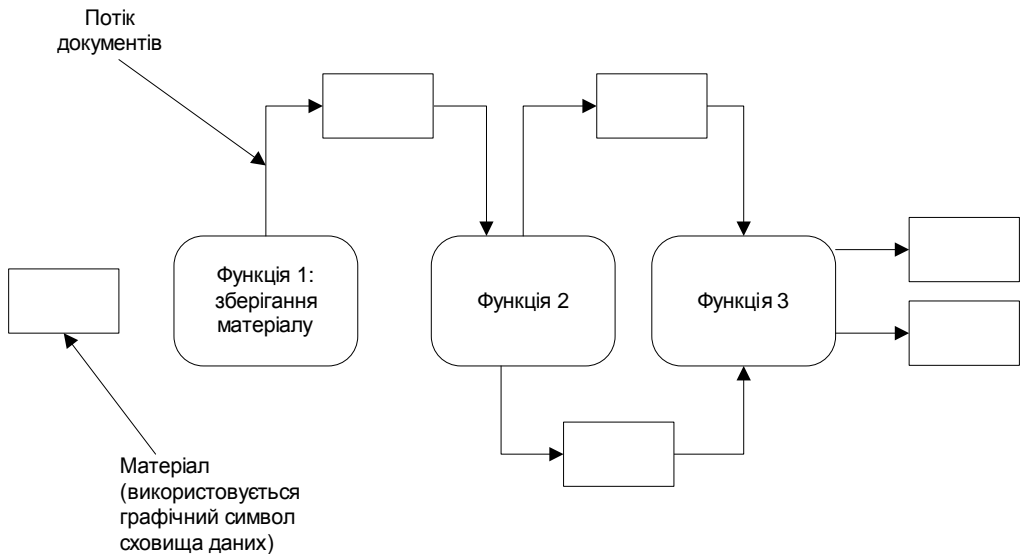


Рис. 4.34. Модель процесу в нотації DFD

– виділення основних і допоміжних бізнес-процесів організації. Слід зазначити, що нотація DFD може бути ефективно застосована для опису потоків документів або потоків матеріальних ресурсів. На рис. 4.35 показаний приклад застосування нотації DFD для цих цілей.



Варіант використання 1



Варіант використання 2

Рис. 4.35. Опис потоків документів (варіант 1) або потоків матеріальних ресурсів (варіант 2)

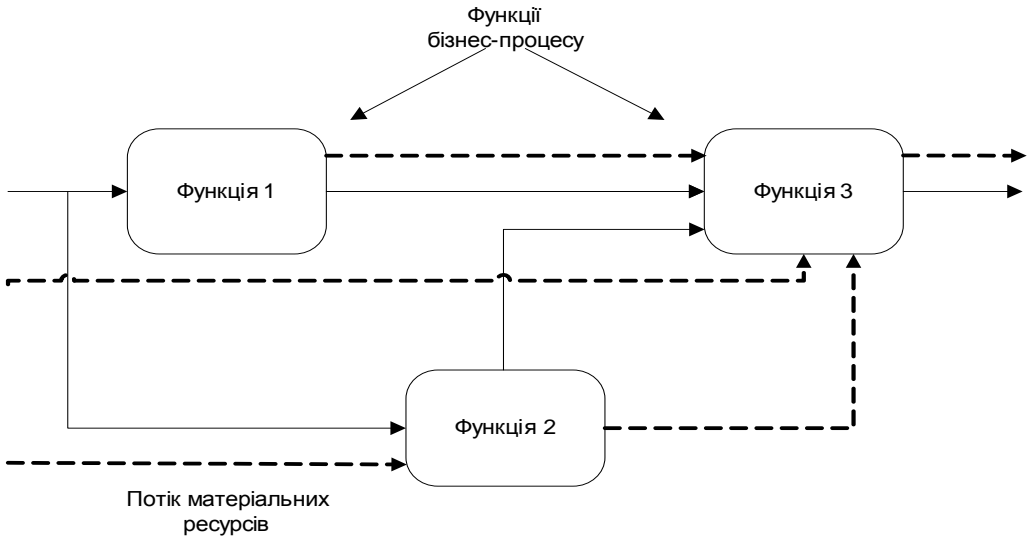


Рис. 4.36. Поєднання різних типів стрілок на одній моделі DFD

Більш того, нотація DFD може бути декілька модернізована так, щоб на одній діаграмі можна було б показати як потоки даних, так і потоки матеріальних ресурсів (рис. 4.36).

На практиці при створенні моделей процесів часто буває корисно використовувати декілька способів опису. Спочатку, наприклад, ми створюємо модель у нотації IDEF0, виявляємо функції, що входять у процес. Потім проводимо декомпозицію процесу. Досягши деякого рівня деталізації (тричотири), доцільно сформулювати для кожного детального процесу декілька схем в різних форматах: управління – IDEF0, а потоки даних і матеріалів – у DFD (див. практикум).

## 4.5. Контрольні запитання, завдання і тести

### 4.5.1. Контрольні запитання

1. Поясніть, в чому полягає смисл діаграми «сутність-зв'язок».
2. Які моделі входять в методологію IDEF?
3. Розшифруйте значення стрілок ICOM.
4. Опишіть правила створення моделі IDEF0.
5. Поясніть різницю між моделями IDEF0, DFD, IDEF3.
6. Опишіть сутність моделі DFD.
7. Чим відрізняються ресурси процесу, що моделюються стрілками зліва і знизу на моделі IDEF0?
8. Яка мінімальна кількість стрілок може бути присутня на контекстній діаграмі процесу в моделі IDEF0?

9. Поясніть правила злиття і розгалуження стрілок на моделі IDEF0.
10. Що таке декомпозиція моделі?
11. Які логічні оператори використовуються в моделі IDEF3?

#### **4.5.2. Завдання для практичних робіт**

1. Розробіть модель контекстного рівня бізнес-процесу прийому співробітника на роботу в організацію.
2. Змодельуйте бізнес-процес в нотації IDEF0 (2 рівні), в якому Ви особисто приймаєте участь.
3. Розробіть діаграму потоків даних для бізнес-процесу, власником якого Ви є.
4. Змодельуйте бізнес-процес закупівель в компанії в нотації DFD (2 рівні).
5. Розробіть модель технологічного процесу виробу будь-якого продукту, причому з використанням нотації IDEF0 на двох рівнях і IDEF3 на третьому рівні декомпозиції.

#### **4.5.3. Питання для самостійної роботи**

1. Історія розвитку методів структурного системного аналізу.
2. Історія появи методології моделювання SADT.
3. Стрілки в тунелях на моделі IDEF0.
4. Типи зворотніх зв'язків на моделі IDEF0.
5. Переваги та недоліки моделі IDEF0.
6. Поясніть, в яких випадках використовується модель IDEF, в яких – IDEF3?
7. Типи стрілок на моделі DFD.
8. Основні елементи нотації DFD.

#### **4.5.4. Тести для самоконтролю**

1. Нотація IDEF0 призначена для:
  - моделювання потоків даних;
  - моделювання потоків робіт;
  - моделювання різних сценаріїв виконання робіт.
2. Стрілка «механізм» на моделі IDEF0 означає:
  - ресурс, що споживається процесом;
  - результат процесу;
  - ресурс, що забезпечує виконання процесу, кількість якого не зменшується в процесі.
3. Сховище даних в нотації DFD – це:
  - спосіб зберігання інформації;
  - матеріальний об'єкт або фізична особа, які можуть виступати як джерело або приймач інформації;

- сукупність операцій щодо перетворення вхідних потоків даних на вихідні.
4. Стрілки, що входять в процес згори на нотації IDEF0, означають:
    - управління процесом;
    - інформаційні ресурси процесу;
    - виконавці процесу.
  5. Декомпозиція процесу – це:
    - злиття кількох функцій в одну;
    - деталізація процесу на нижньому рівні;
    - розгалуження стрілки.
  6. Коли стрілка на нотації IDEF0 береться в тунель:
    - коли вона не підписана;
    - коли вона розгалужується;
    - коли вона відсутня на моделі верхнього рівня.
  7. Стрілки в нотації DFD означають:
    - матеріальні потоки;
    - інформаційні потоки;
    - взаємозв'язок функцій.