

Лекція 4

МЕТОДОЛОГІЯ МОДЕЛЮВАННЯ IDEF0. ОСНОВНІ ПОЛОЖЕННЯ

4.1 Принципи функціонального моделювання

Світ, у якому ми живемо, можна розглядати як складну взаємозалежну сукупність природних і штучних систем. Це можуть бути досить складні системи (наприклад, планети в складі Сонячної системи), системи середньої складності (космічний корабель) або надскладні системи (системи молекулярних взаємодій у живих організмах). Існує величезна кількість наукових дисциплін, призначених для вивчення і пояснення різних аспектів цього нескінченного спектра складності. Наприклад, механіка може пояснити гравітаційне притягання двох планет, а хімія може описати молекулярні взаємодії в склянці окропу. Штучні системи по своїй складності, як правило, займають середнє положення. Наприклад, всесвітня телефонна мережа містить десятки або навіть сотні тисяч перемикачів, однак кількість взаємодій цих перемикачів не витримує ніякого порівняння з кількістю взаємодій молекул навіть у невеликій склянці води. З погляду загальної теорії систем такі системи звичайно розглядаються як системи середньої складності.

Під терміном «моделювання» ми розуміємо процес створення точного опису системи. Особливо важким виявляється опис систем середньої складності, таких, як система комутацій у телефонних мережах, керування повітряними перевезеннями або рухом підводного човна, складання автомобілів, човникові космічні рейси, функціонування переробних підприємств. З погляду людини, ці системи описати досить важко, тому що вони настільки великі, що практично неможливо перелічити всі їхні компоненти зі своїми взаємозв'язками, і в той же час недостатньо великі для застосування загальних припущень, що спрощують (як це прийнято у фізиці). Наша нездатність дати простий опис, а отже, і забезпечити розуміння таких систем робить їхнє проектування і створення трудомістким і дорогим процесом і підвищує ступінь їхньої ненадійності. З ростом технічного прогресу адекватний опис систем стає усе більше актуальною проблемою.

Опис системи за допомогою IDEF0 називається моделлю. В IDEF0 моделях використовуються як природна, так і графічна мови. Для передачі інформації про конкретну систему джерелом природної мови служать люди, що описують систему, а джерелом графічної мови – сама методологія IDEF0. Графічна мова IDEF0 забезпечує структуру і точну семантику природній мові моделі. Графічна мова IDEF0 організує природну мову цілком певним і

однозначним чином, за рахунок чого IDEF0 і дозволяє описувати системи, які донедавна не піддавалися адекватному поданню.

З погляду IDEF0 модель може бути зосереджена або на функціях системи, або на її об'єктах. IDEF0-моделі, орієнтовані на функції, прийнято називати функціональними моделями, а орієнтовані на об'єкти системи – моделями даних, функціональна модель представляє з необхідним ступенем деталізації систему функцій, які у свою чергу відбивають свої взаємовідносини через об'єкти системи. Моделі даних є дуальними до функціональних моделей і являють собою докладний опис об'єктів системи, зв'язаних системними функціями.

4.1.1 Модель відповідає на питання

IDEF0-модель дає повний, точний і адекватний опис системи, що має конкретне призначення. Це призначення, назване метою моделі, впливає з формального визначення моделі в IDEF0:

М є модель системи S, якщо М може бути використана для одержання відповідей на питання відносно S з точністю А.

Таким чином, метою моделі є одержання відповідей на деяку сукупність питань. Ці питання неявно присутні (маються на увазі) у процесі аналізу й, отже, вони керують створенням моделі і направляють його. Це означає, що сама модель повинна буде дати відповіді на ці питання із заданим ступенем точності. Якщо модель відповідає не на всі питання або її відповіді недостатньо точні, то ми говоримо, що модель не досягла своєї мети.

Тільки зрозумівши, наскільки добре потрібно відповісти на поставлені питання, можна визначити, коли процес моделювання можна вважати завершеним (тобто коли модель буде відповідати поставленій меті).

4.1.2 Модель має єдиний суб'єкт

Модель є деяким тлумаченням системи. Тому суб'єктом моделювання служить сама система. Однак модельована система ніколи не існує ізольовано: вона завжди пов'язана з навколишнім середовищем. Причому найчастіше важко сказати, де закінчується система і починається середовище. Із цієї причини в методології IDEF0 підкреслюється необхідність точного визначення границь системи. IDEF0-модель завжди обмежує свій суб'єкт, тобто модель установлює точно, що є і що не є суб'єктом моделювання, описуючи те, що входить у систему, і маючи на увазі те, що лежить за її межами. Обмежуючи суб'єкт, IDEF0-модель допомагає сконцентрувати увагу саме на описуваній системі й дозволяє уникнути включення сторонніх суб'єктів.

4.1.3 У моделі може бути тільки одна точка зору

З визначенням моделі тісно зв'язана позиція, з якої спостерігається система і створюється її модель. Оскільки якість опису системи різко знижується, якщо він не сфокусований ні на чому, IDEF0 вимагає, щоб модель розглядалася увесь час із однієї і тієї ж позиції. Ця позиція називається «точкою зору» даної моделі.

«Точку зору» найкраще уявляти собі як місце (позицію) людини або об'єкта, на яке треба стати, щоб побачити систему в дії. Із цієї фіксованої точки зору можна створити погоджений опис системи так, щоб модель не дрейфувала навколо та біля, і в ній не змішувалися б незв'язані описи.

Іноді тільки одна з багатьох можливих точок зору може дати опис, що задовольняє мети моделі. Наприклад, для створення погодженої моделі механічного цеху можна встати на точку зору як майстра, так і механіка або контролера, але жодна з них сама по собі не дасть моделі, що дозволила б написати навчальний посібник для всього персоналу. Тільки з позиції начальника цеху можна побачити всі види робіт, виконуваних у цеху.

4.1.4 Моделі як взаємозалежні набори діаграм

Після того як визначено суб'єкт, мету і точку зору моделі, починається перша інтеграція процесу моделювання за методологією IDEF0. Суб'єкт визначає, що включити в модель, а що виключити з неї. Точка зору диктує авторові моделі вибір потрібної інформації про суб'єкт і форму її подачі. Мета стає критерієм закінчення моделювання. Кінцевим результатом цього процесу є набір ретельно взаємопов'язаних описів, починаючи з опису самого верхнього рівня всієї системи й кінчаючи докладним описом деталей або операцій системи.

Кожен з таких ретельно взаємопогоджуваних описів називається діаграмою. Модель IDEF0 поєднує і організує діаграми в ієрархічні структури, у яких діаграми верхніх рівнів моделі менш деталізовані, чим діаграми нижніх рівнів. Інакше кажучи, модель IDEF0 можна представити у вигляді деревоподібної структури діаграм, де верхня діаграма є найбільш загальною, а самі нижні найбільш деталізовані. На рисунку 4.1 представлені дві діаграми з моделі експериментального механічного цеху. Верхня діаграма (на вершині моделі) описує механічний цех як функцію, в основі якої лежить перетворення вхідних робочих комплектів (заготівель, сировини, документації) у деталі при певному контролі якості. Нижня діаграма деталізує верхню, указуючи на три головні функції механічного цеху: керування виконанням завдань, виконання завдання і контроль якості виконання. Це приклад того, як IDEF0 організує опис системи, створюючи ієрархію деталей, що додаються на кожному рівні.

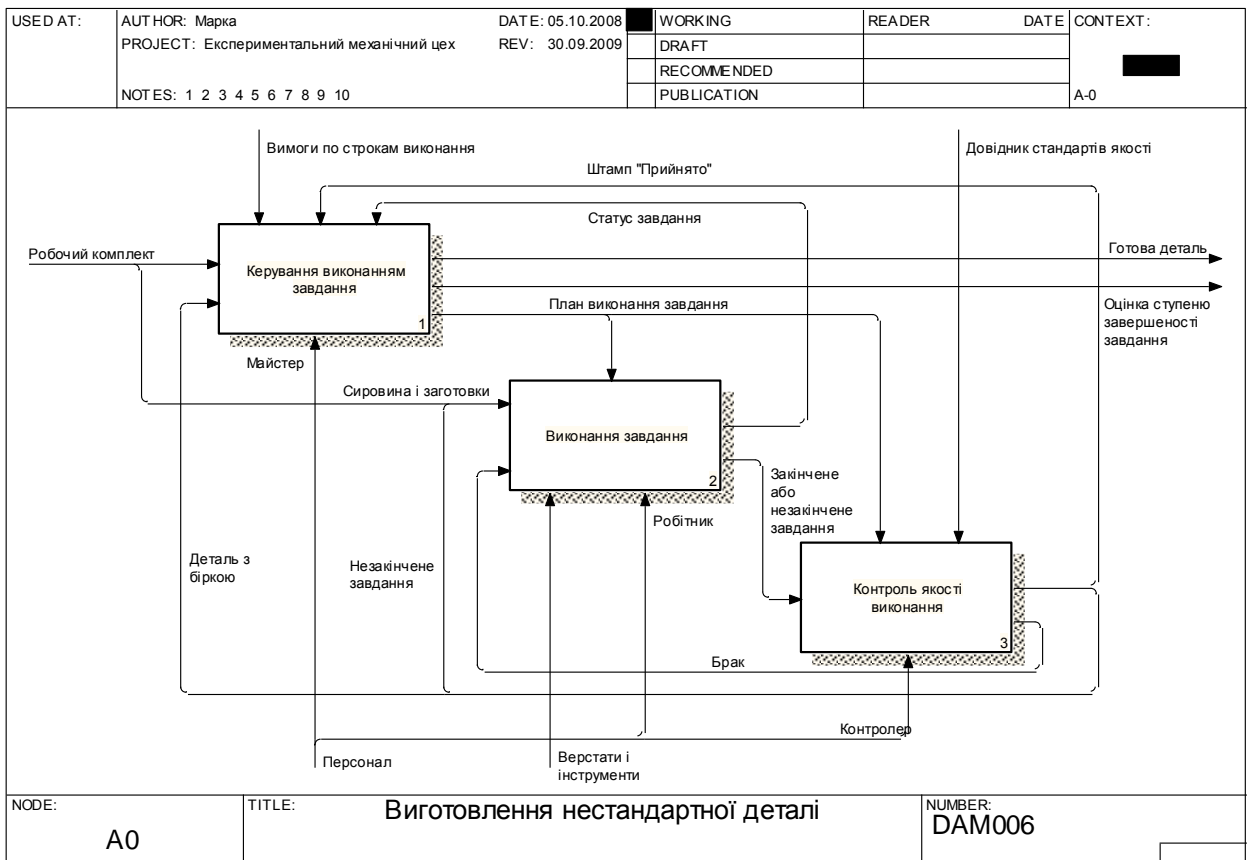
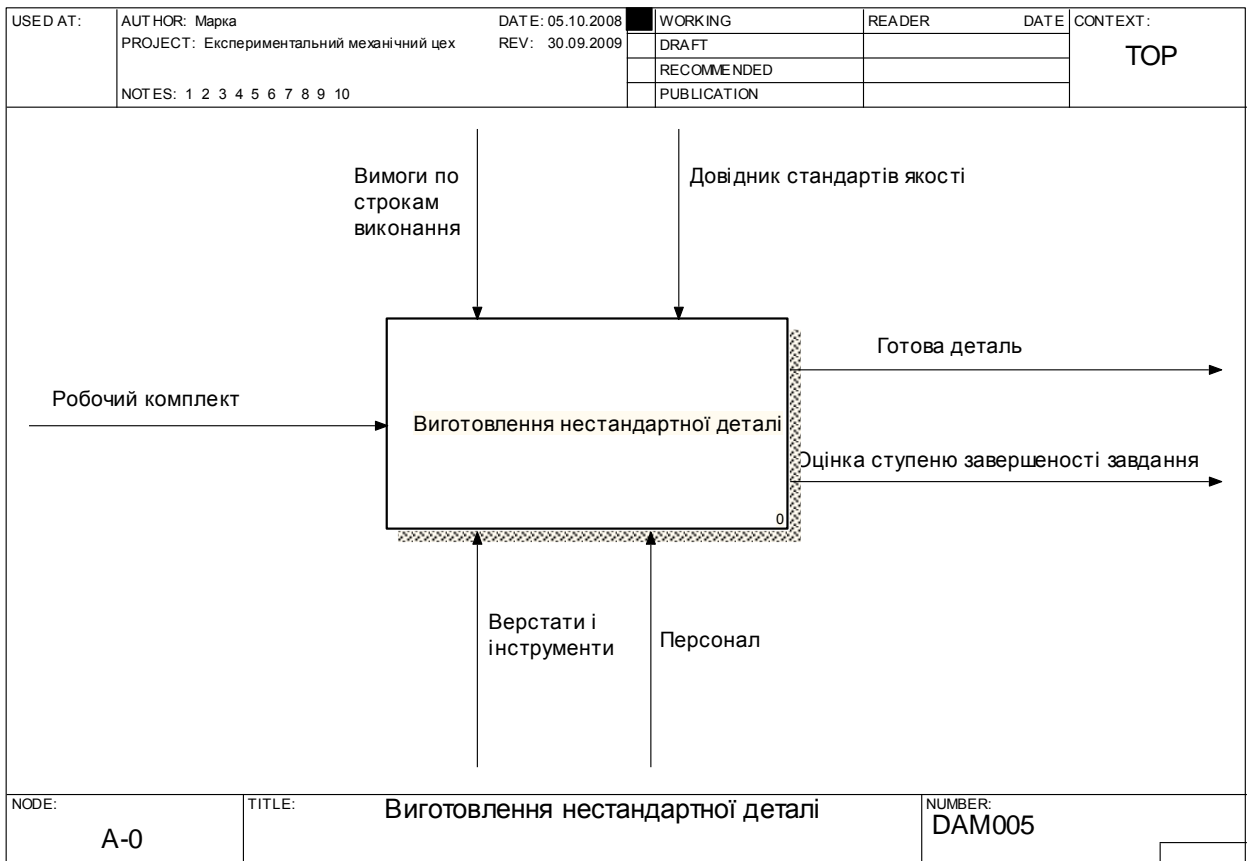


Рисунок 4.1 – Дві взаємозв'язані діаграми IDEF0

На рисунку показано також взаємний вплив трьох функцій нижньої діаграми, позначений дугами, які символізують об'єкти механічного цеху. Якщо уважно подивитися на діаграму, то можна помітити, що деякі дуги доходять до її границі, а імена цих дуг збігаються з тими, що зазначено на дугах верхньої діаграми. Це приклад того, як IDEF0 з'єднує діаграми в моделі через об'єкти системи. Така схема з'єднання вимагає погодженого найменування і обліку об'єктів системи для того, щоб дві діаграми можна було розглядати як зв'язані між собою. Наприклад, функціональний блок на верхній діаграмі має сім дуг, і кожна з них може бути знайдена серед дуг, що йдуть до границі або від границі діаграми на наступному рівні.

4.2 Синтаксис опису діаграм

Діаграма є основним робочим елементом при створенні моделі. Розробник діаграм і моделей звичайно називається аналітиком, або, у термінології IDEF0, автором. Діаграми мають власні синтаксичні правила, що відрізняються від синтаксичних правил моделей. Важливо їх добре розуміти, оскільки графічні позначення мають особливий смисл. Графіка IDEF0 дозволяє визначити різні системні функції і показати, як функції впливають одна на одну.

4.2.1 Діаграми містять роботи і стрілки

Кожна IDEF0-діаграма містить прямокутні блоки, які називаються роботами (activity), і стрілки (arrow). Роботи зображують процеси та функції модельованої системи. Стрілки зв'язують блоки разом і відображають взаємодії і взаємозв'язки між ними (рис. 4.1). Діаграмі дається назва, що розташовується в центрі нижньої частини її бланка. На кожній діаграмі написана стандартно ідентифікуюча її інформація: автор діаграми, частиною якого проекту є робота, дата створення або останнього перегляду діаграми, статус діаграми. Вся ідентифікуюча інформація розташовується у верхній частині бланка діаграми.

4.2.2 Роботи представляють функції

Функціональні блоки на діаграмах зображуються прямокутниками. Робота представляє функцію або активну частину системи, тому назвами робіт служать дієслова або дієслівні звороти. Наприклад, назвами блоків діаграми *“Виготовлення нестандартної деталі”* є: *“Керування виконанням завдання”*, *“Виконання завдання”*, *“Контроль якості виконання”*.

Крім того, IDEF0 вимагає, щоб на діаграмі було не менш трьох і не більше шести блоків (принцип обмеження складності). Ці обмеження

підтримують складність діаграм і моделі на рівні, доступному для читання, розуміння і використання, забезпечують їхню наочність.

Кожна сторона роботи має особливе, цілком певне призначення. Ліва сторона роботи призначена для входів, верхня – для управління, права – для виходів, нижня – для механізмів (рис. 4.2). Таке позначення відбиває певні системні принципи: входи перетворюються у виходи, керування обмежує або приписує умови виконання перетворень, механізми показують ресурси, які потрібні для виконання функції.

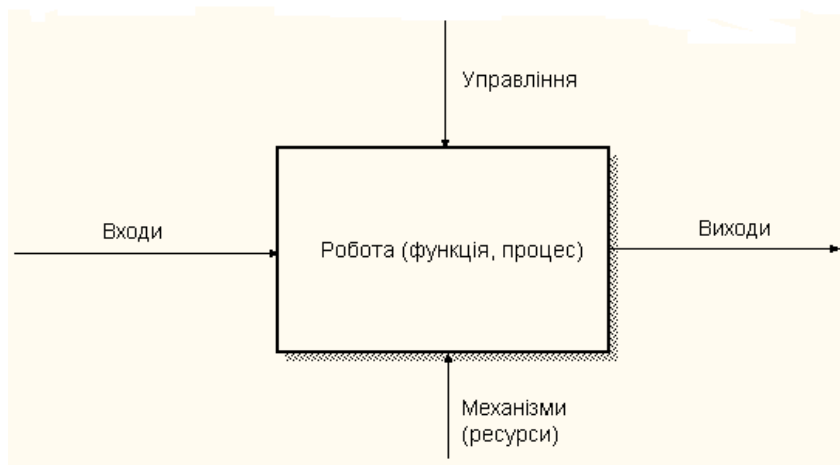


Рисунок 4.2 – Позначення роботи на діаграмі IDEF0

4.2.3 Роботи мають домінування

Роботи ніколи не розміщуються на діаграмі випадковим чином. Вони розміщаються за ступенем важливості, як її розуміє автор діаграми. Цей відносний порядок називається домінуванням. Домінування розуміється як вплив, який одна робота здійснює на інші роботи діаграми. Наприклад, найбільш домінуючою роботою діаграми може бути або перша з необхідної послідовності функцій, або плануюча або контролююча функція, що впливає на всі інші функції (така, як «Керування виконанням завдання» на рис. 4.1).

Найбільш домінуюча робота зазвичай розміщається у верхньому лівому куті діаграми, а найменш домінуюча – у правому нижньому куті. У результаті виходить «східчаста» схема, як на рис. 4.1.

Розташування робіт на сторінці відбиває авторське визначення домінування. Таким чином, топологія діаграми показує, які функції впливають на інші. Щоб підкреслити це, аналітик може перенумерувати роботи відповідно до порядку їхнього домінування. Порядок домінування може позначатися цифрою, розміщеною в правому нижньому куті кожного прямокутника: 1 буде вказувати на найбільше домінування, 2 – на наступне після найбільшого, і т.д.

Роботи в IDEF0 повинні бути перенумеровані. Номера робіт служать однозначними ідентифікаторами для системних функцій і автоматично організують ці функції в ієрархію моделі. Використовуючи номери робіт і оцінюючи вплив, який одна робота здійснює на іншу, аналітик може організувати модель за принципом функціонального домінування. Це дозволяє погодити ієрархічний порядок функцій у моделі з рівнем впливу кожної функції на іншу частину системи. Тому настійно рекомендується в міру можливості нумерувати роботи відповідно до порядку їхнього домінування.

4.2.4 Стрілки зображують об'єкти

Для функціональних діаграм стрілка може представляти багато об'єктів, наприклад, плани, дані в комп'ютерах, машини, інструменти, устаткування і т.д. Однак у системному аналізі замість терміна «об'єкти» часто вживають термін «дані». Це пояснюється тим, що системному аналізу раніше піддавалися, як правило, системи програмного забезпечення.

Оскільки стрілки зображують об'єкти, вони описуються (позначаються) іменниками або іменниками з визначеннями, що розташовуються досить близько до лінії стрілки. Настійно рекомендується розміщати описи стрілок, називані мітками, якнайближче до ліній стрілок, не порушуючи, однак, читабельність діаграм. Це усуває невизначеність у тому, якій дузі належить мітка, і виключається необхідність у додаткових графічних зв'язках. Рекомендується прийняти цей стиль опису стрілок для того, щоб діаграми були впорядкованими і простими для читання.

4.2.5 Стрілки зображують взаємозв'язки між роботами

Між об'єктами і функціями можливі чотири відношення: вхід, управління, вихід, механізм. Кожне із цих відношень зображується стрілкою, пов'язаною з певною стороною роботи. За згодою ліва сторона роботи призначена для вхідних дуг, верхня сторона – для управлінських дуг, права сторона – для вихідних дуг, нижня сторона – для дуг механізмів. Таким чином, сторони роботи чисто графічно сортують об'єкти, зображувані дотичними до роботи стрілками.

Розрізняють чотири типи стрілок:

1) вхід (input) – матеріал або інформація, які використовуються або перетворюються функцією для одержання результату (виходу). Стрілка входу завжди входить в ліву грань роботи. При моделюванні інформаційних систем, коли стрілками є не фізичні об'єкти, а дані, не все так очевидно. Наприклад, при «Прийомі пацієнта» карта пацієнта може бути і на вході і на виході, тоді як

якість цих даних змінюється. Тому стрілки входу і 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 управлінням. У цьому випадку підказкою може служити те, переробляються (змінюються) дані в роботі чи ні. Якщо змінюються, то, швидше за все, це вхід, якщо ні – управління;

2) управління (control) – правила, стратегії, процедури або стандарти, якими керується робота. Кожна робота повинна мати хоча б одну стрілку управління. Управління 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 (управління або контроль) рекомендується рисувати стрілку управління;

3) вихід (output) – матеріал або інформація, які виробляються роботою. Кожна робота повинна мати хоча б одну стрілку виходу. Робота без результату не має змісту і не повинна моделюватися;

4) механізм (mechanism) – ресурси, які виконують роботу, наприклад персонал підприємства, верстати, пристрої і т.д. Стрілка механізму завжди входить в нижню грань роботи. На розсуд аналітика стрілки механізму можуть не зображуватися в моделі.

У методології IDEF0 потрібно тільки п'ять типів взаємозв'язків між роботами для опису їхніх відношень: управління, вхід, зворотний зв'язок по управлінню, зворотний зв'язок по входу, вихід-механізм.

Зв'язок по входу (output-input), коли стрілка виходу вище розташованої роботи направляє на вхід нижче розташованої роботи, наприклад, на рис. 4.3 стрілка «Деталі» зв'язує роботи «Виготовлення деталей» і «Складання виробу».

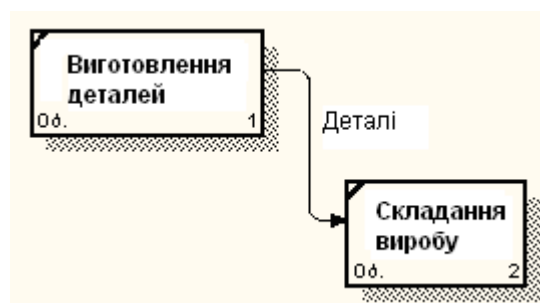


Рисунок 4.3 – Зв'язок по входу

Зв'язок по управлінню (output-control), коли вихід вище розташованої роботи направляється на управління нижче розташованої роботи. Зв'язок по управлінню показує домінування вище розташованої роботи. Дані або об'єкти виходу вище розташованої роботи не змінюються в нижче розташованій. На рис. 4.4 дуга «Креслення» зв'язує роботи «Створення кресленні деталі» і «Виготовлення деталей», при цьому креслення не змінюється у процесі виготовлення деталей.

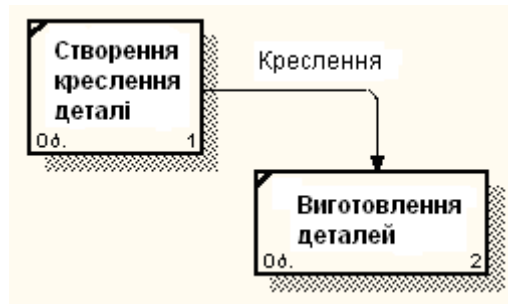


Рисунок 4.4 – Зв'язок по управлінню

Зворотний зв'язок по входу (output-input feedback), коли вихід нижче розташованої роботи направляється на вхід вище розташованої роботи. Такий зв'язок, як правило, використовується для опису циклів. На рис. 4.5 стрілка «Брак» зв'язує роботи «Переробка сировини» і «Контроль якості», при цьому виявлений на контролі брак направляється на вторинну переробку.



Рисунок 4.5 – Зворотний зв'язок по входу

Зворотний зв'язок по управлінню (output-control feedback), коли вихід нижче розташованої роботи направляється на управління вище розташованої роботи (стрілка «Рекомендації», рис. 4.6). Зворотний зв'язок по управлінню часто свідчить про ефективність бізнесу-процесу. На рис. 4.6 якість виробу може бути підвищена шляхом безпосереднього керування процесами

виготовлення деталей і складання виробу залежно від результату (виходу) роботи «Контроль якості».

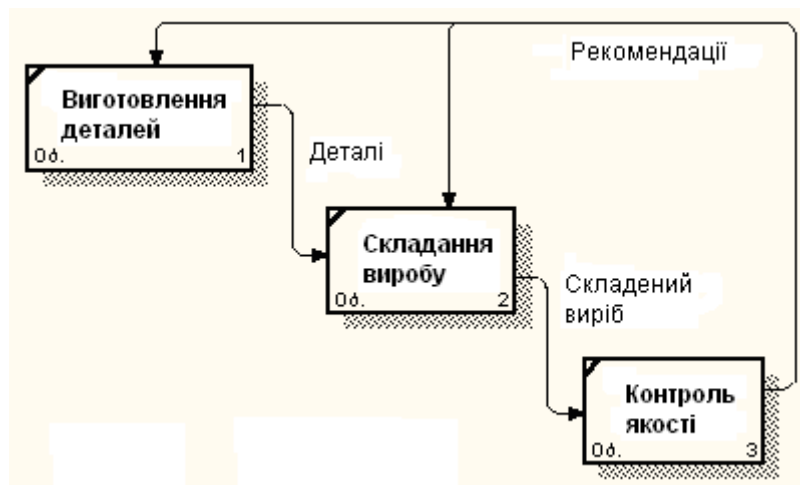


Рисунок 4.6 – Зворотний зв’язок по управлінню

Зв’язок вихід-механізм (output-mechanism), коли вихід однієї роботи направляється на механізм іншої. Цей взаємозв’язок використовується рідше ніж інші і показує, що одна робота підготовляє ресурси, необхідні для проведення іншої роботи (рис. 4.7).

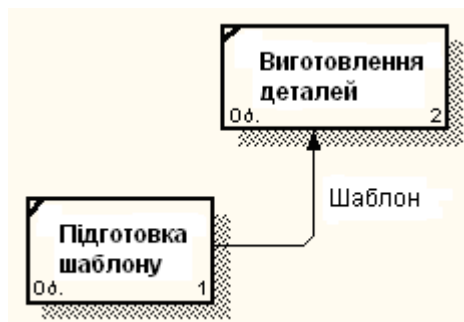


Рисунок 4.7 – Зв’язок вихід-механізм

Зв’язки вихід-механізм характерні при розподілі джерел ресурсів (наприклад, необхідні інструменти, навчений персонал, фізичний простір, устаткування, фінансування, матеріали).

4.2.6 Стрілки представляють набори об’єктів

Стрілка в IDEF0 рідко зображує один об’єкт. Зазвичай вона символізує набір об’єктів. Наприклад, на рис. 4.1 стрілка, іменована «Робочий комплект»,

відбиває «Технічне завдання», «Креслення», «План-графік», деякі «Сировину і заготовки». Оскільки стрілки представляють набори об'єктів, вони можуть мати багато початкових точок (джерел) і кінцевих точок (призначень). Тому стрілки можуть розгалужуватися і з'єднуватися різними складними способами. Вся стрілка або її частина може виходити з однієї або декількох робіт і закінчуватися в одній або декількох роботах, як, наприклад, стрілка «Прийняте завдання» на рис. 4.1. Відзначимо, як різні компоненти стрілки «Прийняте завдання» входять в інші роботи діаграми: «Штамп "Прийнято"» є управляючою інформацією для роботи «Керування виконанням завдання», у той час як «Прийняте, але незакінчене завдання» є входом у роботу «Виконання завдання».

Цей приклад ілюструє, як гілки стрілок показують, із чого складається набір об'єктів. На рис. 4.1 видно також, як «Робочий комплект» і «Прийняте, але незакінчене завдання» з'єднуються разом, створюючи «Сировину і заготовки» – вхід для роботи «Виконання завдання». У цьому випадку дві стрілки об'єднуються, створюючи більший набір. Для пояснення того, як стрілки представляють роз'єднання і з'єднання наборів об'єктів, в IDEF0 були розроблені спеціальні угоди щодо подання і опису розгалужень і з'єднань стрілок.

Розгалуження стрілок

Розгалуження стрілок, зображувані у вигляді ліній, які розходяться, означають, що весь вміст стрілки або його частина може з'явитися в кожному відгалуженні стрілки. Стрілка завжди позначається до розгалуження, щоб дати назву всьому набору. Крім того, кожна гілка стрілки може бути позначена або не позначена у відповідності з такими правилами:

- непозначені гілки містять всі об'єкти, зазначені в мітці стрілки перед розгалуженням (тобто всі об'єкти належать цим гілкам);
- гілки, позначені після точки розгалуження, містять всі об'єкти або їхню частину, зазначену в мітці стрілки після розгалуження (тобто кожна мітка гілки уточнює, що саме містить гілка).

Зміст стрілок, які розгалужуються і зливаються, передається іменуванням кожної гілки таких стрілок. Якщо стрілка іменована до розгалуження, а після розгалуження жодна з гілок не іменована, то мається на увазі, що кожна гілка моделює ті ж об'єкти, що й гілка до розгалуження (рис. 4.8).

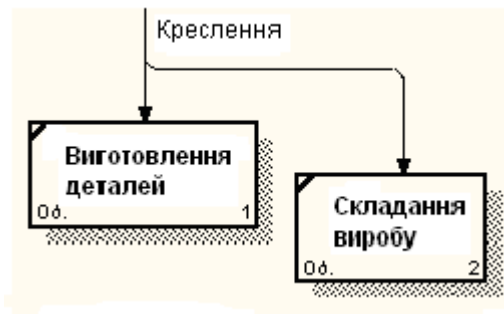


Рисунок 4.8 – Приклад іменування стрілки, що розгалужується

Якщо стрілка іменована до розгалуження, а після розгалуження яка-небудь із гілок не іменована, то мається на увазі, що вона моделює ті ж дані або об'єкти, що й гілка до розгалуження (рис. 4.9).

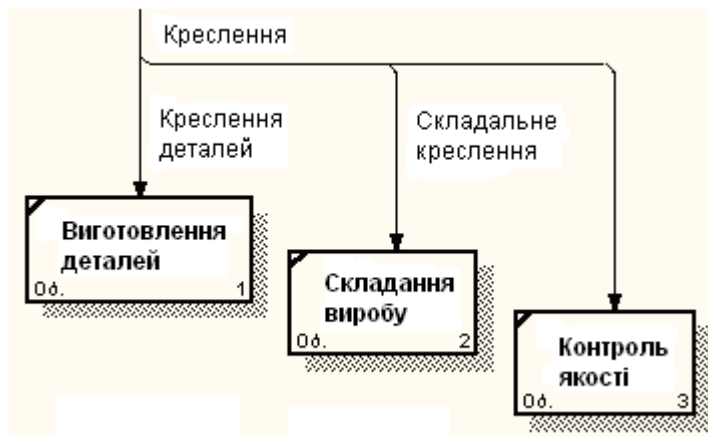


Рисунок 4.9 – Інший приклад іменування стрілки, що розгалужується

Злиття стрілок

Злиття стрілок в IDEF0, зображуване як лінії, що сходяться разом, вказує, що вміст кожної гілки йде на формування мітки для стрілки, що є результатом злиття вихідних стрілок. Після злиття результуюча стрілка завжди позначається для вказівки нового набору об'єктів, що виник після об'єднання. Крім того, кожна гілка перед злиттям може позначатися або не позначатися у відповідності з такими правилами:

- непозначені гілки містять всі об'єкти, зазначені в загальній мітці стрілки після злиття (тобто всі об'єкти виходять із всіх гілок);

- позначені перед злиттям гілки містять всі або деякі об'єкти з перерахованих у загальній мітці після злиття (тобто мітка гілки ясно вказує, що містить гілка).

Наприклад, «Сировина і заготовки» як частина дуги «Робочий комплект» сходяться разом із «Прийнятим, але незакінченим завданням» для формування головного входу у функціональний блок «Виконання завдання». Сировина і заготовки – це назва, що включає і ті й інші об'єкти, тому стрілка після з'єднання одержує цю мітку.

4.3 Ідентифікація версій діаграм С-номерама

При створенні IDEF0-моделі ту саму діаграму разом з її роботами і стрілками перерисовують кілька разів, що приводить до появи різних її варіантів. Щоб розрізнити різні версії однієї й тієї ж діаграми, в IDEF0 використовується схема контролю конфігурації діаграм, заснована на хронологічних номерах, або С-номерах. С-номерні коди утворюються з ініціалів автора і послідовних номерів. Ці коди ставляться в нижньому правому куті бланка діаграми. Наприклад, DAM006 – це С-номер для діаграми «Виготовлення нестандартної деталі» на рис. 4.1. Якщо діаграма заміняє більш старий варіант, то автор поміщає попередній С-номер у дужках, щоб вказати на зв'язок з попередньою роботою. Наприклад, діаграма DAM010 заміняє попередню версію DAM009. Кожний автор проекту IDEF0 веде реєстр всіх створених ним діаграм, нумеруючи їх послідовними цілими числами. Для цього використовується бланк реєстру С-номерів IDEF0.