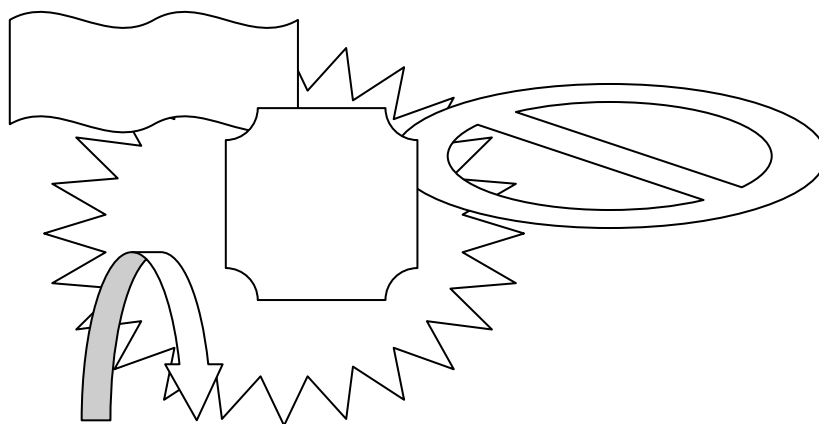


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ М. П. ДРАГОМАНОВА

С. М. Яшанов, С. Б. Дзус

**СТРУКТУРНЕ ТА ІНФОРМАЦІЙНЕ
МОДЕЛЮВАННЯ
В СЕРЕДОВИЩІ DESIGN/IDEF**

Навчально-методичний посібник



Київ
Вид-во НПУ імені М. П. Драгоманова
2017

УДК 378.091.33-027.22:004(076)
ББК 74.025.3р30я7
Я 96

*Друкується за ухвалою Вченої ради
Національного педагогічного університету імені М. П. Драгоманова
(протокол № від червня 2017 р.)*

Рецензенти: *М. І. Жалдак*, академік АПН України, доктор педагогічних наук, професор, завідувач кафедри теоретичної інформатики НПУ імені М. П. Драгоманова;
О. М. Спірін, член-кореспондент НАПН України, доктор педагогічних наук, професор, заступник директора з наукової роботи Інституту інформаційних технологій і засобів навчання НАПН України.

Яшанов С. М.

Я 96

Структурне та інформаційне моделювання в середовищі DESIGN/IDEF: навчально-методичний посібник / С. М. Яшанов, С. Б. Дзус. – К. : Вид-во НПУ імені М. П. Драгоманова, 2017. – 73 с.

У навчально-методичному посібнику представлено теоретичні матеріали та лабораторний практикум з методології структурного аналізу і проектування у середовищі моделювання DESIGN/IDEF для студентів спеціальності 015 «Професійна освіта. Комп'ютерні технології».

Розглянуто методи використання сучасних засобів структурного аналізу і проектування у середовищі моделювання DESIGN/IDEF спрямованих на формування та розвиток у студентів системи інформатичних компетентностей, які забезпечують здатність майбутнього педагога ефективно здійснювати професійну діяльність в галузі структурного моделювання у середовищі DESIGN/IDEF.

Видання розраховано на студентів спеціальності 015 «Професійна освіта. Комп'ютерні технології» та викладачів вищих педагогічних навчальних закладів, інститутів післядипломної педагогічної освіти.

УДК 378.091.33-027.22:004(076)
ББК 74.025.3р30я7

ISBN

© Яшанов С. М., Дзус С. Б., 2017
© Вид-во НПУ імені М. П. Драгоманова, 2017

1. МЕТОДОЛОГІЇ СТРУКТУРНОГО МОДЕЛЮВАННЯ

Проектування інформаційних систем (ІС) - логічно складна, трудомістка і тривала робота, що вимагає високої кваліфікації фахівців-проектантів. У процесі створення і функціонування ІС інформаційні потреби користувачів постійно змінюються або уточнюються, що ще більше ускладнює розробку і супровід таких систем.

У багатьох аспектах системний аналіз та моделювання систем є найбільш важкими частинами розробки. Проблеми, з якими зіштовхується системний аналітик, взаємопов'язані (і це є однією з головних причин їх важкого вирішення):

- аналітику складно отримати вичерпну інформацію для оцінки вимог до системи з точки зору замовника;
- замовник, в свою чергу, не має достатньої інформації про проблему обробки даних, щоб судити, що є здійсненним, а що - ні;
- аналітик зіштовхується з надмірною кількістю докладних відомостей про предметну галузь і про нову систему;
- специфікація системи через значний обсяг технічних термінів часто незрозуміла для замовника;
- в разі зрозумілості специфікації для замовника, вона буде недостатньою для проектувальників і програмістів, що створюють систему.

Ці проблеми можуть бути істотно полегшені за рахунок застосування сучасних структурних методів аналізу та моделювання, серед яких центральне місце займають методології структурного аналізу.

Структурний аналіз - це систематичний покроковий підхід до аналізу вимог і проектування специфікацій системи незалежно від того, чи є вона існуючою або створюється знову.

Методологія структурного аналізу і проектування визначає кроки роботи, які повинні бути виконані, їх послідовність, правила розподілу і призначення операцій і методів.

В наш час успішно використовуються такі методології, як структурний

системний аналіз Гейне-Сарсона, структурний аналіз і проектування Йордана де Марко, SADT і інші.


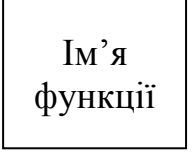
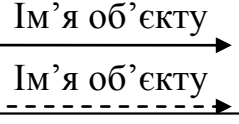
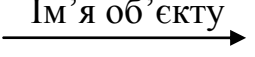
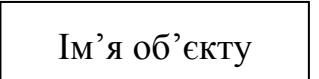
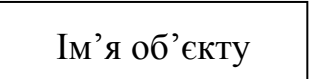
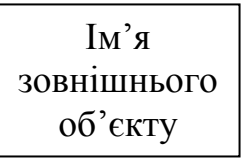
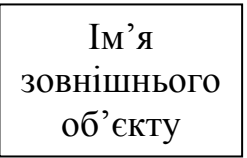
1.1. Діаграми потоків даних. Нотація Йордана-де Марко. Нотація Гейне-Сарсона

Діаграми потоків даних (DFD - Data Flow Diagramm) є основним засобом моделювання функціональних вимог до проектованої системи. З їх допомогою ці вимоги розбиваються на функціональні компоненти (процеси) і представляються у вигляді мережі, пов'язаної потоками даних. Головна мета таких засобів - продемонструвати, як кожен процес перетворює свої вхідні дані у вихідні, а також виявити відношення між цими процесами.

Для зображення DFD традиційно використовуються дві різні нотації: Йордана-де Марко і Гейне-Сарсона. Діаграми потоків даних (DFD - Data Flow Diagramm) будуються з елементів, представлених в таблиці 1.

Таблиця 1.1.

Конструктивні елементи діаграми потоків даних

Елемент	Нотація Йордана-де Марко	Гейна-Сарсона
Функція		
Потік даних		
Сховище даних		
Зовнішня сутність		

На діаграмах функціональні вимоги представляються за допомогою процесів і сховищ, пов'язаних потоками даних.

Потоки даних є механізмами, що використовуються для моделювання передачі інформаційних потоків (або навіть фізичних компонент) з однієї частини системи в іншу. Важливість цього об'єкта очевидна: він дає назву цілому інструменту. Потоки на діаграмах зазвичай зображуються іменованими стрілками, орієнтація яких вказує напрямок руху інформаційних потоків.

Іноді інформаційний потік може рухатися в одному напрямку, оброблятися і повертатися назад в джерело. Така ситуація може моделюватися або двома різними потоками, або одним - двонаправленим.

Призначення *Функції* полягає в продукуванні вихідних потоків із вхідних відповідно до дії, що задається ім'ям функції. Це ім'я має містити дієслово в невизначеній формі з подальшим доповненням (наприклад, ОБЧИСЛИТИ МАКСИМАЛЬНУ ВИСОТУ). Крім того, кожна функція повинна мати унікальний номер для посилань на нього всередині діаграми. Цей номер може використовуватися спільно з номером діаграми для отримання унікального індексу функції у всій моделі.

Сховище дозволяє на певних ділянках визначати дані, які будуть зберігатися в пам'яті між процесами. Фактично сховище представляє «зрізи» потоків даних у часі. Інформація, яку воно містить, може використовуватися в будь-який час після її визначення, при цьому дані можуть вибиратися в будь-якому порядку. Ім'я сховища має ідентифікувати його вміст і бути іменником. У разі, коли потік даних входить або виходить в / із сховища і його структура відповідає структурі сховища, він повинен мати те ж саме ім'я, яке немає необхідності відображати на діаграмі.

Зовнішня сутність представляє сутність поза контекстом системи, що є джерелом або приймачем системних даних. Її ім'я повинно містити іменник, наприклад, СКЛАД ТОВАРІВ. Передбачається, що об'єкти, які представлені такими вузлами, не повинні брати участь ні в якій обробці.

Функції, сховища і зовнішні сутності на DFD-діаграми пов'язуються дугами, що представляють потоки даних. Дуги можуть розгалужуватися або

зливатися, що означає, відповідно, поділ потоку даних на частини, або злиття об'єктів. При інтерпретації DFD-діаграми використовуються наступні правила:

- функції перетворюють вхідні потоки даних у вихідні;
- сховища даних не змінюють потоки даних, а служать тільки для зберігання об'єктів, що надходять у систему;
- перетворення потоків даних у зовнішніх сутностях ігноруються.

Крім цього, для кожного інформаційного потоку і сховища визначаються пов'язані з ними елементи даних. Кожному елементу даних присвоюється ім'я, також для нього може бути визначений тип даних і формат. Саме ця інформація є вихідною на наступному етапі проектування - побудові моделі «сутність-зв'язок». При цьому, як правило, інформаційні сховища перетворюються в сутності, а проектувальнику залишається тільки вирішити питання з використанням елементів даних, які пов'язані зі сховищами.

Приклад 1. На рис. 1.1 представлена DFD-діаграма для підприємства, що будує свою діяльність за принципом «виготовлення на замовлення».

На підставі отриманих замовлень формується план випуску продукції на певний період. Відповідно до цього плану визначаються потреба в комплектуючих виробках і матеріалах, а також графік завантаження виробничого обладнання. Після виготовлення продукції і проведення платежів, готова продукція відправляється замовнику. Ця діаграма являє самий верхній рівень функціональної моделі. Природно, це дуже грубий опис предметної галузі. Уточнення моделі проводиться шляхом деталізації необхідних функцій на DFD-діаграмі наступного рівня. Таким чином можна розбити функцію «визначення потреб і забезпечення матеріалами» на підфункції «визначення потреб», «пошук постачальників», «укладання і аналіз договорів на поставку», «контроль платежів», «контроль поставок», пов'язані власними потоками даних, які будуть представлені на окремій діаграмі. Деталізація моделі повинна проводитися до тих пір, поки вона не

буде містити всю інформацію, необхідну для побудови інфосистеми.

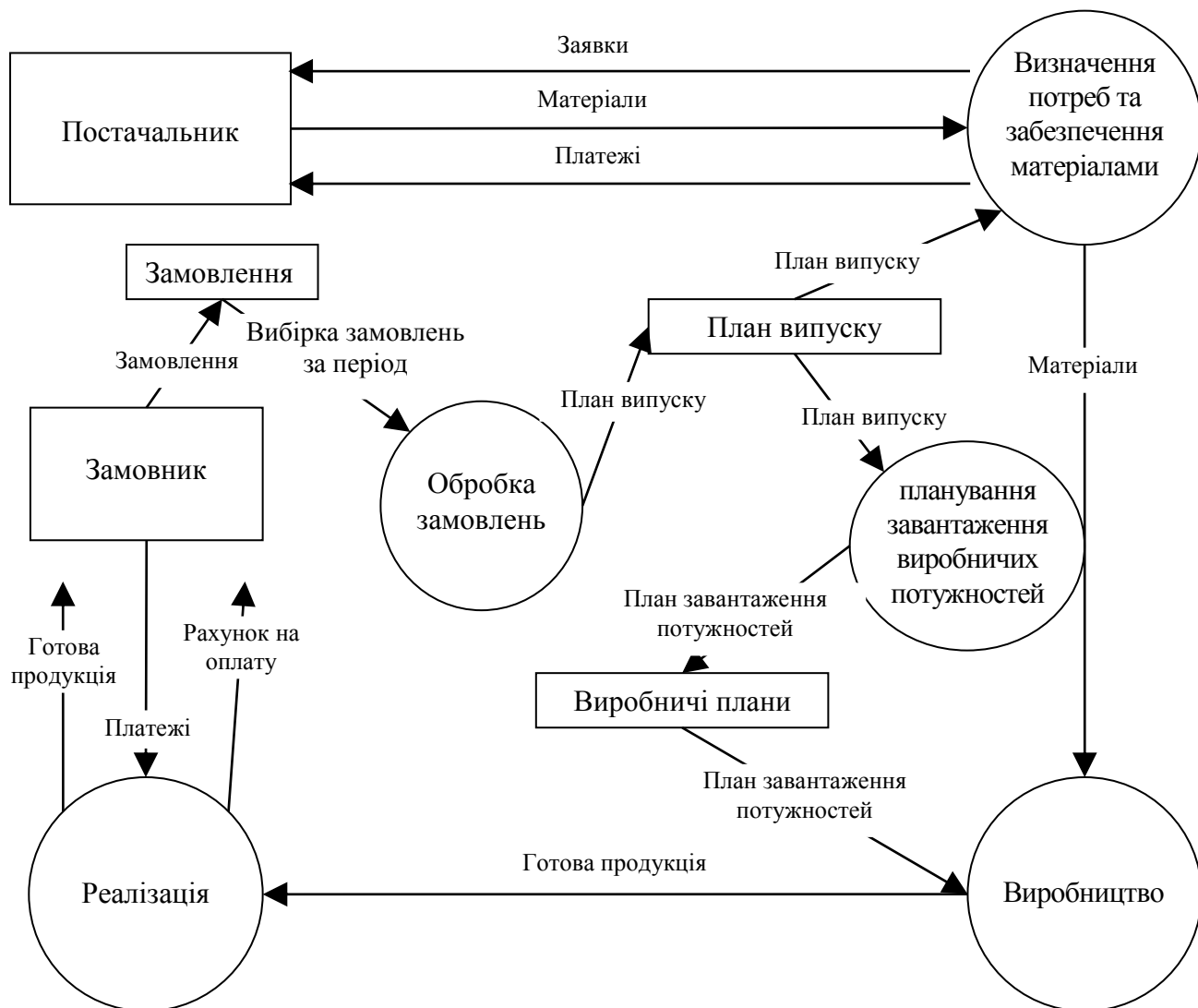


Рис. 1.1. DFD-діаграма виробництва

Приклад 2. Розглянемо процес ЗДАТИ ІСПИТ. У нас є дві сутності СТУДЕНТ і ВИКЛАДАЧ. Наведемо потоки даних, якими обмінюється наша проєктована система з зовнішніми об'єктами (рис. 1.2).

Із сторони сутності СТУДЕНТ опишемо інформаційні потоки. Для складання іспиту необхідно, щоб у СТУДЕНТА була ЗАЛІКОВА КНИЖКА, а також щоб він мав допуск ДО ІСПИТУ. Результатом складання іспиту, тобто вихідними потоками будуть ОЦІНКА ЗА ІСПИТ і ЗАЛІКОВА КНИЖКА, в яку буде виставлена ОЦІНКА.

З боку сутності ВИКЛАДАЧ інформаційні потоки мають наступний вигляд: ЕКЗАМЕНАЦІЙНА ВІДОМІСТЬ, згідно з якою буде відомо, що

СТУДЕНТ допущений до іспиту, а також офіційний лист, куди буде занесений результат іспиту, тобто ОЦІНКА ЗА ІСПИТ, виставлена в ЕКЗАМЕНАЦІЙНІЙ ВІДОМОСТІ.

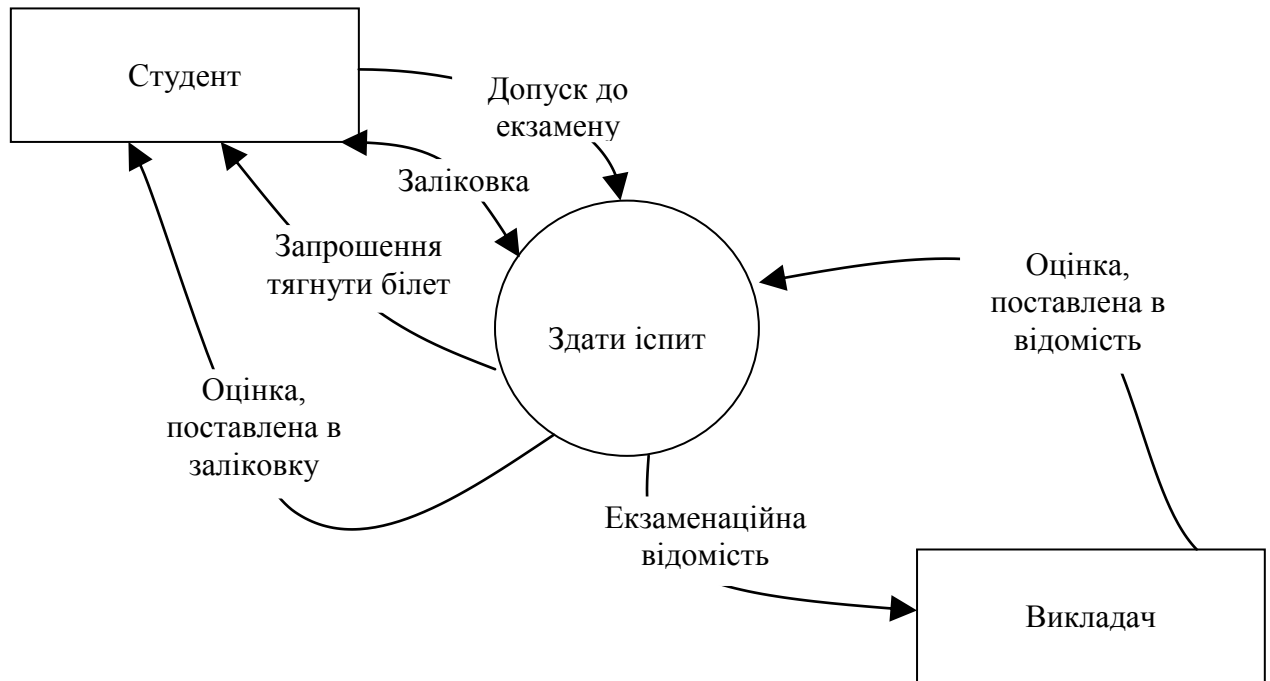


Рис. 1.2. DFD-діаграма процесу «Здати іспит»

Тепер деталізуємо процес ЗДАЧА ІСПИТУ.

Цей процес буде містити наступні процеси (рис. 1.3):

ВИТЯГНУТИ БІЛЕТ {1.1},

ПІДГОТУВАТИСЯ ДО ВІДПОВІДІ {1.2},

ВІДПОВІСТИ НА БІЛЕТ {1.3},

ВИСТАВЛЕННЯ ОЦІНКИ {1.4}.

Розглянуті методології фокусують увагу на потоках даних, їх головне призначення - створення базованих на графіці документів за функціональними вимогами. Методології підтримуються традиційними нисхідними методами проектування специфікацій і забезпечують один із способів зв'язку між аналітиками, розробниками і користувачами системи.

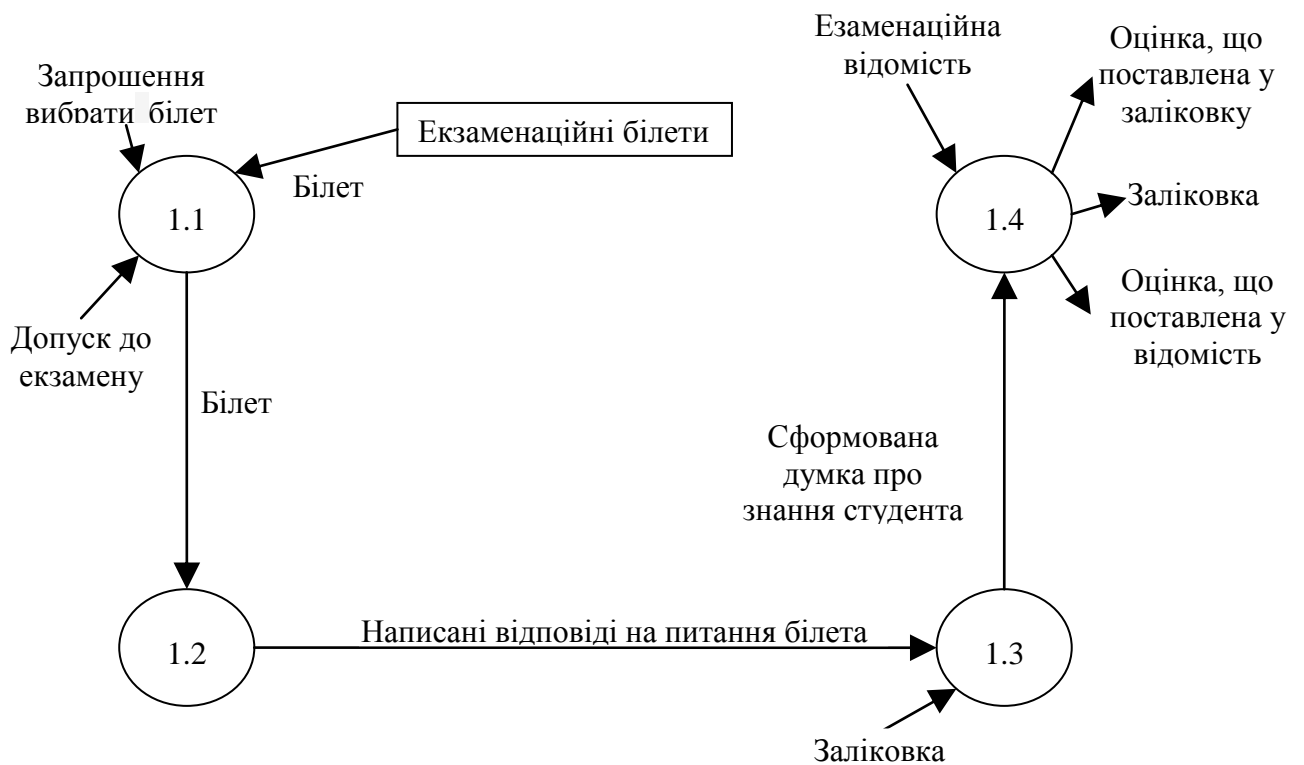


Рис. 1.3. Декомпозиція 1-го рівня DFD-діаграми процесу «Здати іспит»

1.2. Методології IDEF

Для вирішення завдань моделювання складних систем, існують добре обкатані методології та стандарти. До таких стандартів належать методології сімейства IDEF. З їх допомогою можна ефективно відобразити і аналізувати моделі діяльності широкого спектру складних систем в різних розрізах. При цьому широта і глибина обстеження процесів в системі визначається самим розробником, що дозволяє не перевантажувати модель, що створюється, зайвими даними. У табл. 1.2. відображені міжнародні стандарти IDEF.

Таблиця 1.2.

Міжнародні стандарти системи IDEF

Назва	Повна назва	Опис
IDEF0	Function Modeling	Методологія функціонального моделювання. За допомогою наглядної графічної мови IDEF0, система постає перед розробниками та аналітиками у вигляді набору взаємозалежних функцій (функціональних блоків - у термінах IDEF0). Як правило, моделювання засобами IDEF0 є першим етапом вивчення будь-якої системи

IDEF1	Information Modeling	Методологія моделювання інформаційних потоків усередині системи, яка дозволяє відобразити й аналізувати їх структуру і взаємозв'язки
IDEF1X (IDEF1 Extended)	Data Modeling	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 мають відношення до даної системи
IDEF2	Simulation Modeling	Методологія динамічного моделювання розвитку систем. У зв'язку з вельми серйозними складнощами аналізу динамічних систем від цього стандарту практично відмовилися, і його розвиток призупинився на самому початковому етапі. Проте в даний час присутні алгоритми та їх комп'ютерні реалізації, що дозволяють перетворювати набір статичних діаграм IDEF0 у динамічні моделі, побудовані на базі «розфарбованих мереж Петрі» (CPN - Color Petri Nets)
IDEF3	Process Description Capture	Методологія документування процесів, що відбуваються в системі, яка використовується, наприклад, при дослідженні технологічних процесів на підприємствах. За допомогою IDEF3 описуються сценарій та 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	Object-oriented Design	Методологія побудови об'єктно-орієнтованих систем. Засоби IDEF4 дозволяють наочно відобразити структуру об'єктів і закладені принципи їх взаємодії, тим самим дозволяючи аналізувати й оптимізувати складні об'єктно-орієнтовані системи
IDEF5	Ontology Description Capture	Методологія онтологічного дослідження складних систем. У IDEF5 онтологія системи описується за допомогою словника термінів і правил, на підставі яких формуються твердження про стан системи в певний момент часу. На основі цих тверджень формуються висновки про подальший розвиток системи і проводиться її оптимізація

IDEF6	Design Rationale Capture	Обґрунтування проектних дій. Призначення IDEF6 полягає в полегшенні отримання «знань про спосіб» моделювання, їх подання та використання при розробці систем управління підприємствами. Під «знаннями про спосіб» розуміються причини, обставини, приховані мотиви, які обумовлюють вибрані методи моделювання. Простіше кажучи, «знання про спосіб» інтерпретуються як відповідь на питання: «чому модель вийшла такою, якою вийшла?» Більшість методів моделювання фокусуються на власне одержуваних моделях, а не на процесі їх створення. Метод IDEF6 акцентує увагу саме на процесі створення моделі
IDEF7	Information System Audit Method	Аудит інформаційних систем. Цей метод був спочатку затребуваний, проте так і не був повністю розроблений
IDEF8	User Interface Modeling	Стандарт розробки інтерфейсів взаємодії оператора і системи (користувацьких інтерфейсів). Сучасні середовища розробки користувальницьких інтерфейсів більшою мірою створюють зовнішній вигляд інтерфейсу. IDEF8 фокусує увагу розробників інтерфейсу на програмуванні бажаного взаємного поведіння інтерфейсу і користувача на трьох рівнях: <ul style="list-style-type: none"> • виконуваної операції (що це за операція); • сценарії взаємодії, визначеному специфічною роллю користувача (за яким сценарієм вона повинна виконуватися тим чи іншим користувачем); • деталі інтерфейсу (які елементи управління пропонує інтерфейс для виконання операції).
IDEF9	Scenario-driven Info Sys Design Spec	Стандарт дослідження бізнес обмежень для полегшення виявлення та аналізу обмежень в умовах яких діє підприємство. Знання про основні обмеження і характер їх впливу, які закладаються в моделі, в кращому випадку залишаються неповними, неузгодженими, розподіленими нерационально. Це не обов'язково призводить до того, що побудовані моделі нежиттєздатні, а їх реалізація зіткнеться з труднощами. У випадках, коли мова йде саме про вдосконалення структур або адаптації до змін, знання про існуючі обмеження мають критичне значення

IDEF10	Implementation Architecture Modeling	Моделювання архітектури виконання. Цей стандарт так і не був повністю розроблений
IDEF11	Information Artifact Modeling	Цей стандарт так і не був повністю розроблений
IDEF12	Organization Modeling	Організаційне моделювання. Цей стандарт так і не був повністю розроблений
IDEF13	Three Schema Mapping Design	Трисхемне проектування перетворення даних. Цей стандарт так і не був повністю розроблений
IDEF14	Network Design	Стандарт проектування комп'ютерних мереж, заснований на аналізі вимог, специфічних мережевих компонентів, існуючих конфігурацій мереж. Також він забезпечує підтримку рішень, пов'язаних з раціональним управлінням матеріальними ресурсами, що дозволяє досягти суттєвої економії

Методологія SADT (Structured Analysis and Design Technique) розроблена Дугласом Т. Россом у 1969-73 роках. Вона спочатку створювалася для проектування систем загального призначення в порівнянні з іншими структурними методами, що виростили з проектування програмного забезпечення. IDEF0 (підмножина SADT) використовується для моделювання бізнес-процесів в організаційних системах і має розвинені процедури підтримки колективної роботи.

У термінах IDEF0 система представляється у вигляді комбінації блоків і дуг. Блоки представляють функції системи, дуги представляють множини об'єктів (фізичні об'єкти, інформацію або дії, які утворюють зв'язки між функціональними блоками). Місце з'єднання дуги з блоком визначає тип інтерфейсу.

Правила інтерпретації моделі полягають у наступному (рис. 1.4):

- функціональний блок (функція) перетворює вхідні об'єкти у вихідні;
- управління визначає, коли і як це перетворення може або повинно відбутися;
- механізм (виконавець) здійснює це перетворення.



Рис. 1.4. IDEF0-модель підприємства

З дугами зв'язуються мітки на природній мові, що описують дані, які вони представляють. Дуги показують, як функції системи пов'язані між собою, обмінюються даними і здійснюють управління одна з одною. Виходи однієї функції можуть бути входами, управлінням або виконавцями іншої.

Дуги можуть розгалужуватися і з'єднуватися. Розгалуження означає множинність (ідентичні копії одного об'єкта) або розщеплення (різні частини одного об'єкта). З'єднання означає об'єднання або злиття об'єктів.

Кожен блок IDEF0-діаграми може бути представлений декількома блоками, з'єднаними інтерфейсними дугами, на діаграмі наступного рівня. Ці блоки представляють підфункції (підмодулі) вихідної функції. Кожен з підмодулів може бути декомпозований аналогічним чином. Число рівнів не обмежується, проте рекомендується на одній діаграмі використовувати не

менше 3 і не більше 6 блоків.

Приклад 3. На рис. 1.5 представлена IDEF0-модель діяльності підприємства, описаного в прикладі 1.

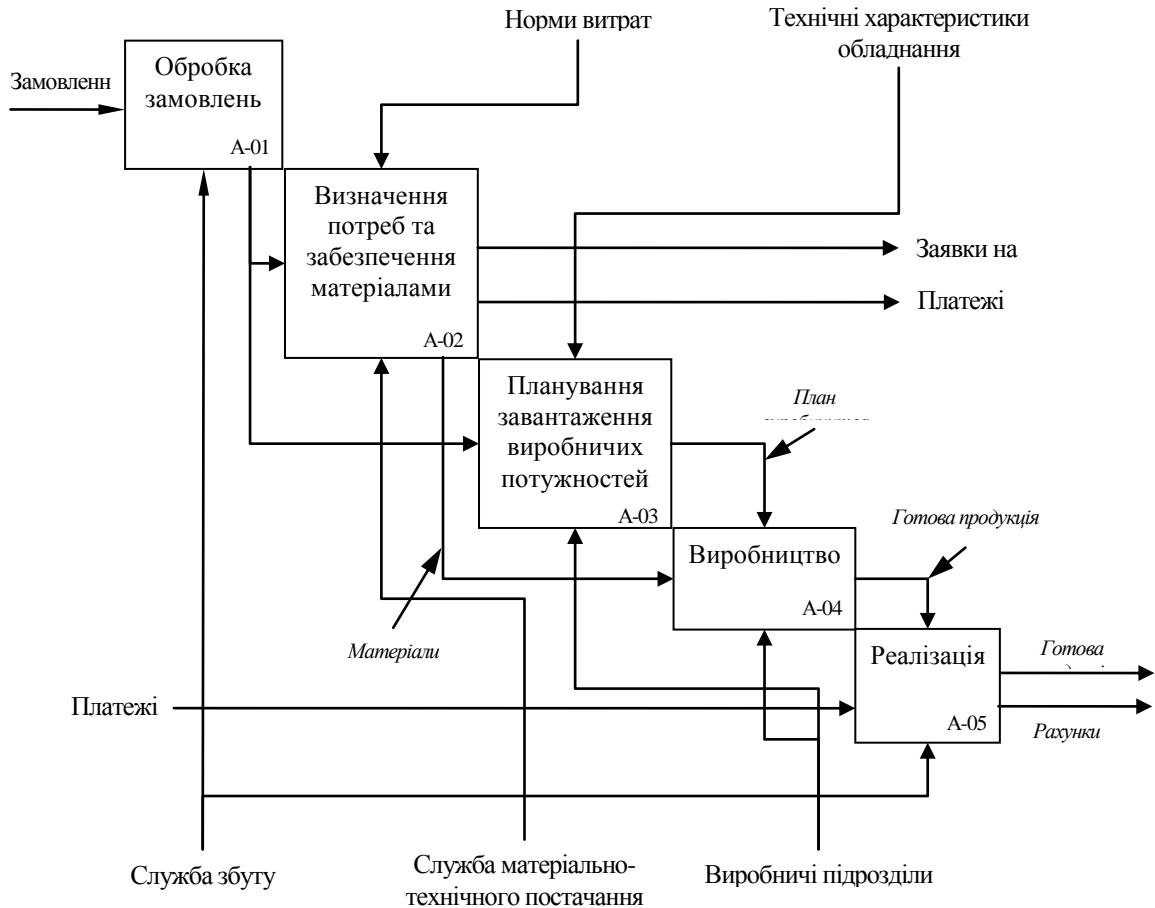


Рис. 1.5. IDEF0-модель підприємства

Методологія SADT може використовуватися для моделювання широкого кола систем і визначення вимог і функцій, а потім для розробки системи, яка задовольняє цим вимогам і реалізує ці функції. Для вже існуючих систем SADT може бути використана для аналізу функцій, що виконуються системою, а також для визначення механізмів, за допомогою яких вони здійснюються.

Результатом застосування методології SADT є модель, яка складається з діаграм, фрагментів текстів і глосарію, що мають посилання один на одного.

2. МЕТОДОЛОГІЇ ІНФОРМАЦІЙНОГО МОДЕЛЮВАННЯ

Найважливіша мета інформаційної моделі полягає у виробленні несуперечливої інтерпретації даних і взаємодій між ними з тим, що необхідно для спільного використання та управління цілісністю даних.

Поява понять концептуальної схеми даних привела до методології семантичного моделювання даних, тобто до визначення значень даних в контексті їх взаємозв'язків з іншими даними.

У 1983 році в рамках проекту військового відомства США «Інтегровані системи інформаційної підтримки» (ICAM) була створена методологія семантичного моделювання даних IDEF1X (розширення методології IDEF1), що дозволяє логічно об'єднувати в мережу неоднорідні обчислювальні системи.

Методологія IDEF1X - один з підходів до семантичного моделювання даних, заснований на концепції Сутність-Зв'язок (Entity-Relationship), це інструмент для аналізу інформаційної структури систем різної природи.

Методологія IDEF1X призначена для побудови концептуальної схеми реляційної бази даних, яка була б незалежною від програмної платформи її кінцевої реалізації.

Ця інформація є необхідним доповненням функціональної IDEFO-моделі, деталізує об'єкти, якими маніпулюють функції системи.

Концептуально IDEF1X-модель можна розглядати як проект логічної схеми бази даних для проєктованої системи.

IDEF1X використовує поняття сутностей, атрибутів, відношень і ключів. Мови графічного зображення моделей, що використовуються цими методологіями, також багато в чому схожі. Однак, IDEF1X не розглядає об'єкти реального світу, а лише їх інформаційне відображення, так як до моменту розробки бази даних всі інформаційні ресурси організації повинні бути вивчені, необхідний набір даних для відображення її діяльності визначений і перевірений на повноту. Оскільки IDEF1X призначена для розробки реляційних баз даних, вона додатково оперує множиною понять,

правил і обмежень, такими як домени, представлення, первинні, зовнішні і сурогатні ключі і іншими, які прийшли з реляційної алгебри і в яких немає необхідності на етапах вивчення та опису діяльності організації.

Стандарт і методологія IDEF1X є спеціалізованим інструментом, призначеним для розробників реляційних баз даних.

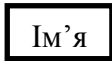
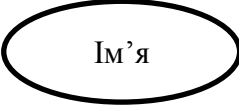
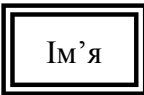

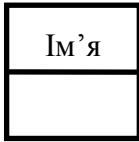
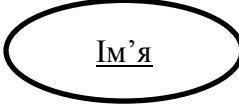
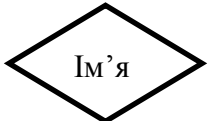
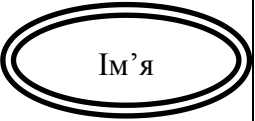

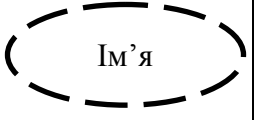
Найбільшого поширення набули такі нотації, що використовуються при побудові ER-діаграм: нотація Чена, нотація Мартіна, нотація IDEF1X, нотація Баркера.

2.1. Нотація Чена

У таблиці 2.1 наведені конструктивні елементи концептуальної схеми бази даних в нотації Чена.

Таблиця 2.1.

Конструктивні елементи нотації Чена

№	Елемент діаграми	Позначення	№	Елемент діаграми	Позначення
1		незалежна сутність	6		атрибут
2		залежна сутність	7		первинний ключ
3		родина сутність в ієрархічному зв'язку	8		зовнішній ключ (поняття зовнішнього ключа вводиться в реляційній моделі даних)
4		зв'язок	9		багатозначний атрибут
5		ідентифікуючий зв'язок	10		одержуваний (успадкований) атрибут в ієрархічних зв'язках

Зв'язок з'єднується з асоційованими сутностями лініями. Біля кожної сутності на лінії, що з'єднує її зі зв'язком, цифрами вказується клас приналежності (рис. 2.1).

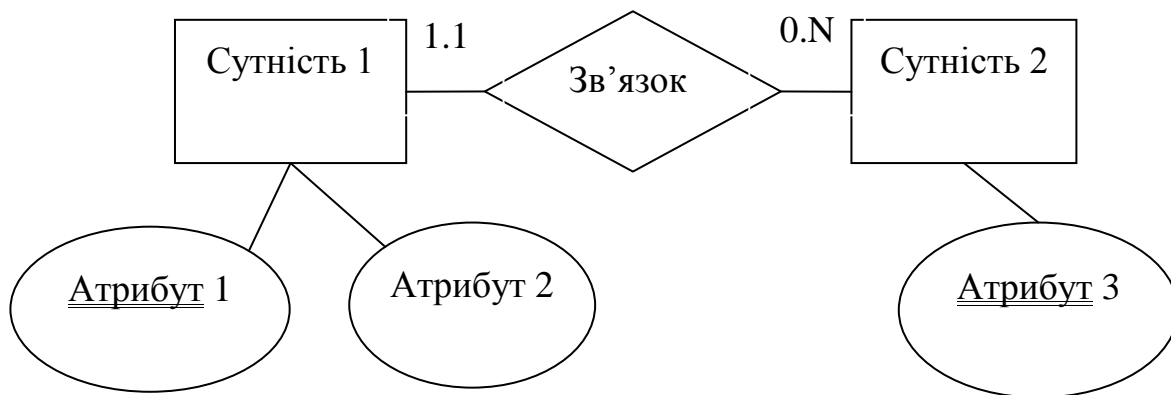


Рис. 2.1. Приклад фрагмента концептуальної схеми в нотації Чена

2.2. Ногація Мартіна

У таблиці 2.2 наведені конструктивні елементи концептуальної схеми бази даних в нотації Мартіна.

Таблиця 2.2.

Конструктивні елементи нотації Мартіна

Елемент діаграми	Позначення
Ім'я	незалежна сутність
Ім'я	залежна сутність
Ім'я	батьківська сутність в ієрархічному зв'язку

Список атрибутів наводиться всередині прямокутника, що позначає сутність. Ключові атрибути підкреслюються. Відношення зображують лініями, що з'єднують сутності, вид лінії в місці з'єднання з сутністю визначає кардинальність зв'язку (таблиця 2.3):

Таблиця 2.3.

Позначення зв'язків в нотації Мартіна

№	Позначення	Карди- нальність	№	Позначення	Карди- нальність
1		Немає	4		M,N
2		1,1	5		0,N
3		0,1	6		1,N

Ім'я зв'язку вказується на лінії, що її позначає (рис. 2.2).

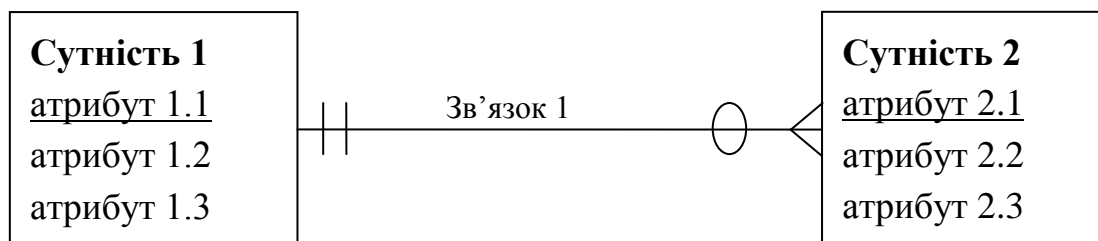


Рис. 2.2. Приклад фрагмента концептуальної схеми в нотації Мартіна

2.3. Нотація IDEF1X

Позначення конструктивних елементів концептуальної схеми бази даних нотації IDEF1X наведені в табл. 2.4 – 2.6.

Таблиця 2.4.

Позначення сутностей в нотації IDEF1X

Елемент діаграми	Позначення
	незалежна сутність
	залежна сутність

Таблиця 2.5.

Позначення зв'язків в нотації IDEF1X

Елемент діаграми	Позначення
	Ідентифікуючий зв'язок
	Не ідентифікуючий зв'язок

Позначення відношення неповної категоризації (сутності-категорії складають неповну множину нащадків загальної сутності) наведено на рис. 2.5.

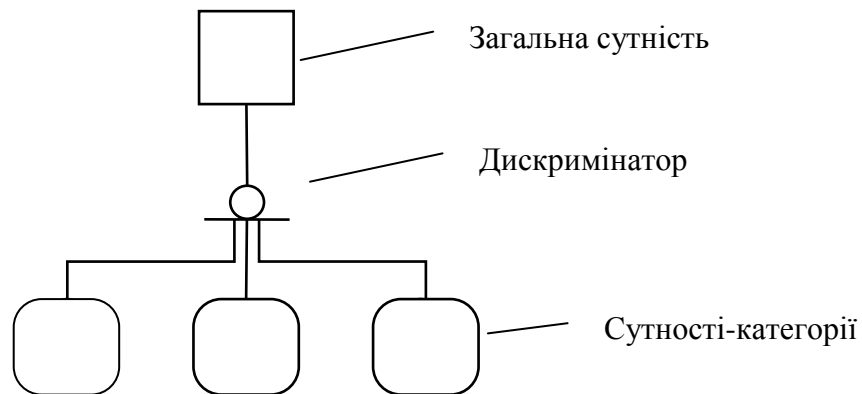


Рис. 2.5. Відношення неповної категоризації в нотації IDEF1X

2.4. Нотація Баркера

Сутності позначаються прямокутниками, всередині яких наводиться список атрибутів. Ключові атрибути відзначаються символом # (решітка). Зв'язки позначаються лініями з іменами (рис. 2.6), місце з'єднання зв'язку і сутності визначає кардинальність зв'язку (таблиця 2.7).

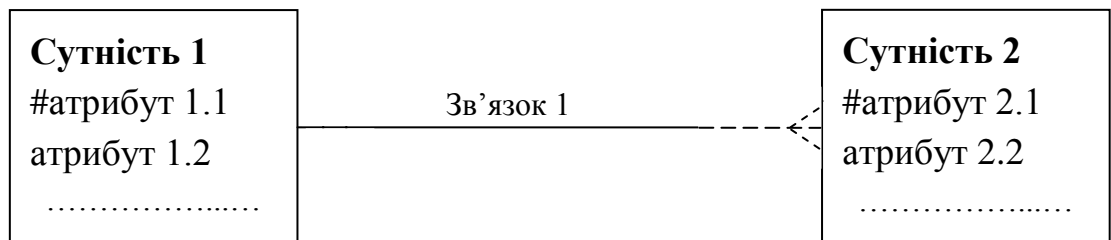


Рис. 2.6. Приклад фрагмента концептуальної схеми в нотації Баркера

Таблиця 2.7.

Позначення зв'язків в нотації Баркера

Позначення	Кардинальність
-----	0,1
————	1,1
-----<-----	0,N
————<-----	1,N

Для позначення відношення категоризації вводиться елемент «дуга» (рис. 2.7).

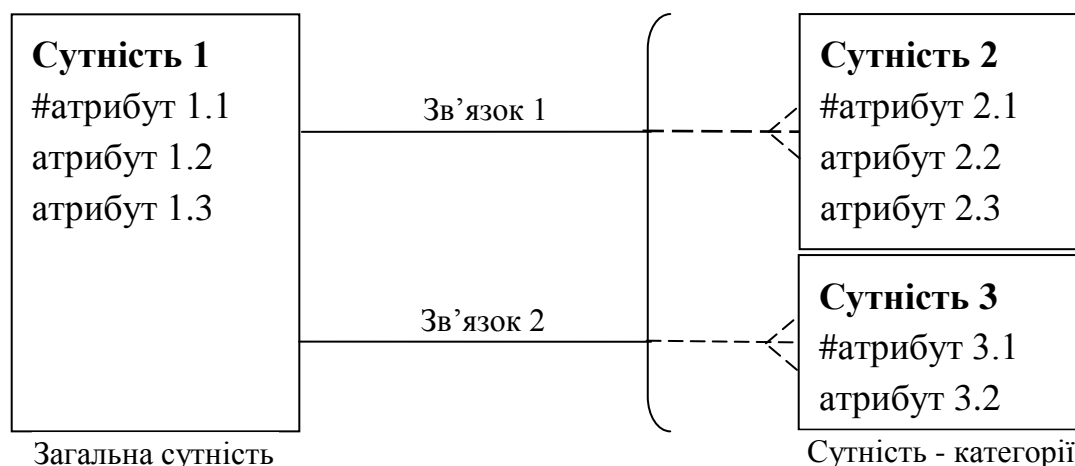


Рис. 2.7. Позначення категоризації в нотації Баркера

3.5. Мова інфологічного моделювання

Мова ER-діаграм використовується для побудови невеликих моделей і ілюстрації окремих фрагментів великих. Найчастіше ж застосовується менш наочна, але більш змістовна мова концептуального моделювання (МКМ), в якій сутності і зв'язки представляються у наступному вигляді:

СУТНІСТЬ (атрибут 1, атрибут 2, ..., атрибут n)

ЗВ'ЯЗОК [СУТНІСТЬ S1, СУТНІСТЬ S2, ...]

(Атрибут 1, атрибут 2, ..., атрибут n),

де S - ступінь зв'язку, а атрибути, що входять у ключ, повинні бути відзначені за допомогою підкреслення.

Так, розглянутий вище приклад множини зв'язків між сутностями (рис. 2.8), може бути описаний на МКМ наступним чином:

Лікар (Номер лікаря, Прізвище, Ім'я, По батькові, Спеціальність)

Пацієнт (Реєстраційний номер, Номер ліжка, Прізвище, Ім'я, По батькові, Адреса, Дата народження, Стать)

Лікуючий лікар [Лікар (1,1), Пацієнт (1, M)]

(Номер лікаря, Реєстраційний номер)

Консультант [Лікар (1, M), Пацієнт (1, N)]

(Номер лікаря, Реєстраційний номер).

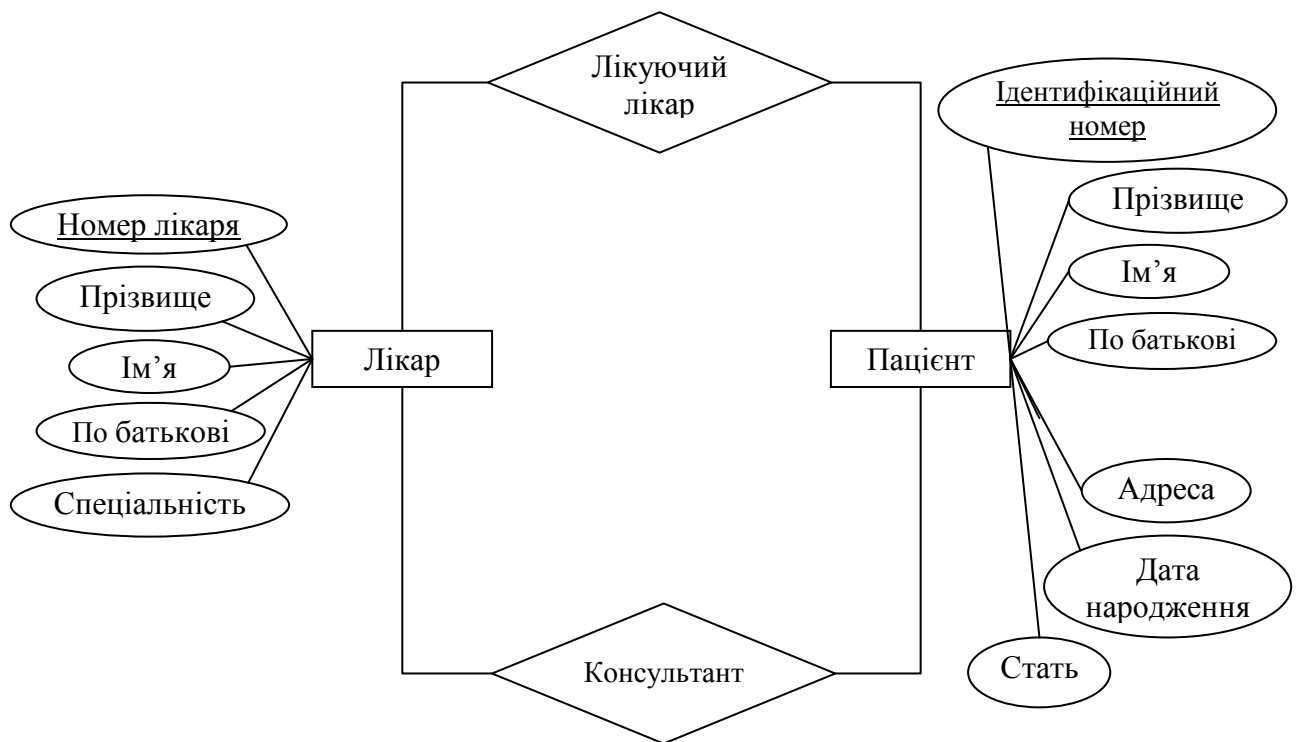


Рис. 2.8. Фрагмент концептуальної схеми в нотації Чена

3. РОЗРОБКА ФУНКЦІОНАЛЬНОЇ МОДЕЛІ ПРЕДМЕТНОЇ ГАЛУЗІ В DESIGN/IDEF

Загальна схема складання функціональної моделі складається з наступних етапів:

- 1) побудова моделі (розробка функціональної діаграми; заповнення глосарію додатковими визначеннями; доповнення діаграми гіпертекстом);
- 2) перевірка синтаксису моделі (перевірка на наявність зв'язків, на ідентифікатори функцій і зв'язків, на управління).

3.1. Запуск IDEF0 моделювання

Перший крок у використанні Design/IDEF для IDEF0 моделювання - відкриття нової IDEF0-сторінки у новій або існуючій IDEF0- моделі.

3.2. Створення нової IDEF0-моделі

Щоб почати створення нової IDEF0-моделі:

- а) запуснуть *Design/IDEF*;

б) виберіть *New* в меню *File*;

в) в діалоговому вікні «*Select New Page Type*» (рис. 3.1) для створення інформаційної моделі в списку *Methodology* виберіть *IDEF0* і підтвердіть вибір клавішею *OK*;

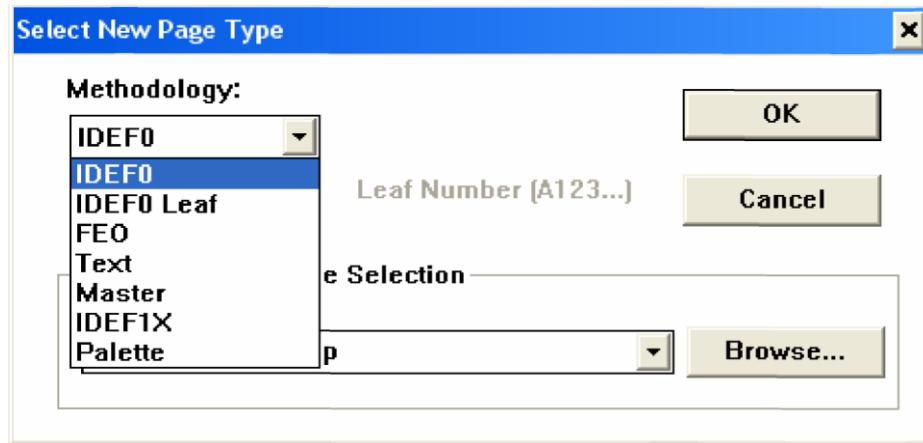


Рис. 3.1. Діалогове вікно «*Select New Page Type*»

г) з'являється чиста IDEF0-сторінка вищого рівня з одним блоком всередині. Це головний об'єкт подальшої роботи. Початкова нумерація для блоку вищого рівня в *Design/IDEF* - «AT». У кожній IDEF0-моделі можна мати тільки одну сторінку вищого рівня.

Створення об'єктів IDEF0 відбувається шляхом виклику пункту *EDEF* *Box* меню *Create* або натисканням кнопки на панелі інструментів (рис. 3.2).



Рис. 3.2. Інструменти для створення IDEF0-моделі

3.3. Налаштування середовища проектування

Перш, ніж створювати інші блоки, бажано відразу налаштувати середовище проектування, так як в іншому випадку в подальшому доведеться встановлювати налаштування для кожного створеного блоку окремо. Для цього в меню Edit, пункті Set Attributes ... необхідно налаштувати всі шрифти, які будуть діяти за замовчуванням. Із запропонованого списку шрифтів слід вибирати кирилицю. Далі в меню Edit, пункті Set Options ... встановлюється максимальна кількість блоків на сторінці (Activities => Maximum Boxes), тип нумерації (Activities => Numbering), кривизну стрілок (Arrows => Automating Routing), валюту і точність обчислень (Cost) і інші (рис. 3.3).

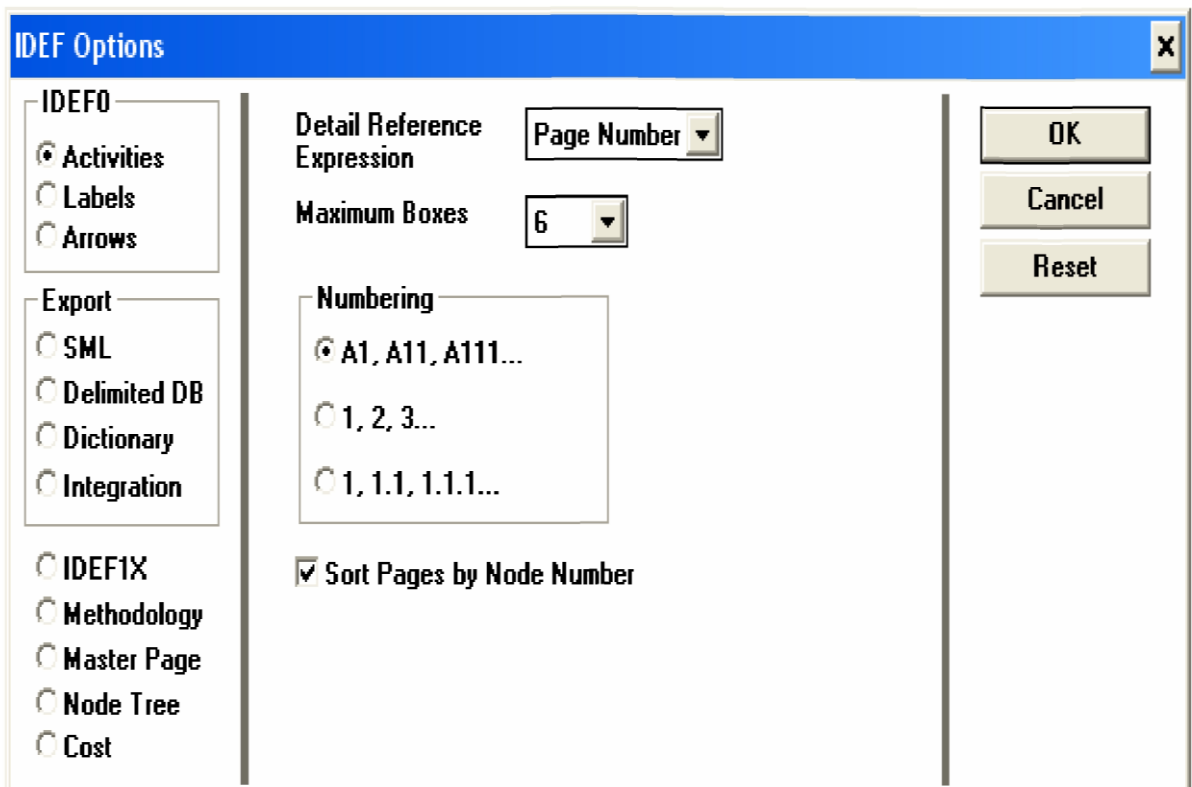



Рис. 3.3. Діалогове вікно «*IDEF Options*»


Також встановлюються налаштування принтера (File => Print Setup ...). Слід встановити альбомне розташування сторінки, а також відповідний формат.

В налаштуваннях атрибутів проекту (Edit => Set Page Attributes ...) встановлюються відповідні атрибути. Також бажано перевірити і інші

налаштування з пунктів Set Colors ..., Set User Options ..., Set Settings ... з меню Edit.

3.4. Створення та редагування функціональних блоків


Для створення нового блоку використовується пункт IDEF Vox меню Create  або відповідна кнопка на панелі інструментів. Після вибору пункту слід вказати місце, де буде розташований новий блок.

Якщо потрібно розмістити кілька блоків на діаграмі, то для цього може бути використаний пункт Place Voxes ... меню Create або кнопка . Після вибору пункту або натискання на кнопку в вікні необхідно буде ввести кількість розміщуваних блоків. Нові блоки розміщуються рівномірно по діагоналі з лівого верхнього в правий нижній кут. Розміри нових блоків встановлюються за замовчуванням. Розміри кожного блоку змінюються шляхом перетягування маркерів (чорних квадратиків на кордоні об'єкта), які з'являються після клацання миші на блоці.

3.5. Додавання тексту до функціонального блоку



Кожен блок у діаграмі повинен бути підписаний. Правильним є іменування блоків за допомогою дієслів невизначеної форми. До початку введення підпису можна заздалегідь встановити потрібні параметри шрифту командою Set Attributes ... пункту меню Edit. Для того, щоб в блоці з'явився курсор, що означає можливість введення імені, необхідно виконати одну з наведених нижче дій:

- відкрити пункт меню Modify, вибрати команду Turn On Text і клацнути всередині функціонального блоку;

- клацнути у лівій панелі інструментів на кнопці  і всередині функціонального блоку.


Режим введення тексту вимикається клавішею Esc.

3.6. Створення та редагування ярликів.

До функціонального блоку з чотирьох сторін повинні бути приєднані різні об'єкти. Для введення назв (ярликів) об'єктів використовується пункт Label меню Create, клавіша F3 клавіатури або кнопка  на панелі інструментів. Після натискання слід вказати місце вставки ярлика і ввести потрібний текст. Скасування режиму введення написів відбувається при натисканні на клавішу Esc або на будь-яку кнопку на панелі інструментів. Редагування ярликів відбувається шляхом включення режиму редагування тексту (Modify => Turn On Text або клавіша F2 або кнопка  на панелі інструментів). Вимкнення відбувається при натисканні клавіші Esc.


3.7. Створення дуг

Дуги на функціональній діаграмі призначені для з'єднання ярликів з функціональними блоками і блоків між собою. щоб з'явилася можливість намалювати дугу, виконуємо один із запропонованих варіантів:

- вибираємо команду Arrow пункту меню Creates
- натискаємо на кнопку  на панелі інструментів.


При появі курсора у вигляді горизонтальної стрілки, розміщуємо його всередині ярлика або функціонального блоку і простягаємо до місця призначення. Щоб проконтролювати правильність з'єднання можна змінити місце положення ярлика. Якщо при цьому дуга тягнеться за ярликом - це означає, що приєднання пройшло успішно.

Дуги на діаграмі можуть зливатися в одну або навпаки роз'єднуватися. Якщо кілька блоків повинні бути з'єднані дугами від одного джерела, чинимо так:

- 1) малюємо дугу до одного з блоків;
- 2) виділяємо її;
- 3) вибираємо команду *Branch* пункту меню Create або кнопку ;

- 4) підводимо курсор до потрібного блоку (він починає блимати);
- 5) клацаємо кнопкою миші, вказуючи на ту сторону даного блоку, до якої повинна підходити дуга.

Якщо кілька дуг від різних блоків повинні з'єднуватися в одну, виконуємо наступну послідовність дій:

- 1) малюємо дугу від першого блоку;
- 2) виділяємо її;
- 3) вибираємо команду *Join* пункту меню *Create* або кнопку ;
- 4) підводимо курсор до наступного блоку (він починає блимати);
- 5) клацаємо кнопкою миші, вказуючи на ту сторону даного блоку, від якої повинна відходити дуга.


Для малювання декількох дуг пункти 2-5 повторюємо потрібну кількість раз.

Якщо до блоку з якогось боку приєднується кілька дуг, бажано розташувати їх рівномірно один щодо одного, що не завжди вдається зробити вручну. Для автоматичної розстановки дуг включите прапорець *Autospace Arrows* в діалоговому вікні, що відкривається послідовністю команд *Edit => Set Attributes ... => Arrow*. Інші установки цього вікна дозволяють вибрати плавне закруглення вигинів дуг, спосіб представлення перетинів дуг і т.п.

3.8. Створення підписів дуг, що з'єднують блоки

Іноді на діаграмі потрібно підписати дугу сполучає функціональні блоки. Ця процедура виконується в два етапи:

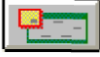
- 1) вводимо підпис поряд з дугою на вільному місці діаграми (команда *Label* в пункті меню *Create*) і потім виділяємо її;

- 2) активізуємо команду *Attach Label* пункту меню *Create* або кнопку  на панелі інструментів і підводимо курсор миші до дузі, яку потрібно підписати. При цьому найближча до курсора дуга починає блимати. Зробити свій остаточний вибір можна, клацнувши лівою кнопкою миші на потрібній дузі. Після закінчення процедури від підпису до дуги буде проведена

хвиляста лінія, що символізує їх взаємний зв'язок.



3.9. Декомпозиція блоків

Для створення такого функціонального рівня моделі помічаємо клацанням миші функціональний блок і виконуємо команду Decompose

пункту меню Create, або вибираємо кнопку  на панелі інструментів. Потім слід двічі клацнути всередині блоку для переходу на нову сторінку моделі.

На даній сторінці моделі вже знаходяться ярлики, створені на попередньому рівні. Крім назв ярликів тут присутні спеціальні позначення (у вигляді сполучення заголовних латинських букв і цифр, наприклад, I1, I2, M1, O1, C1, C2, C3), що походять від англійських слів Input (вхід), Mechanisms (механізм), Output (вихід), Control (управління).

3.10. Переміщення між сторінками IDEF0-моделі

Для переходу на батьківську або дочірню сторінку використовуються пункти Parent і Child меню Select або кнопки  і  на панелі інструментів. Для переходу на довільну сторінку може бути використано ієрархічне дерево сторінок проекту, яке викликається пунктом Page ... меню Select.

3.11. Заповнення глосарію

Перед заповненням глосарію повинні бути задані пункти функціонально-вартісного аналізу. Для цього необхідно викликати пункт Activity Centers ... меню Glossary і у вікні набрати список пунктів, за якими буде проводитися розрахунок (рис. 3.4).

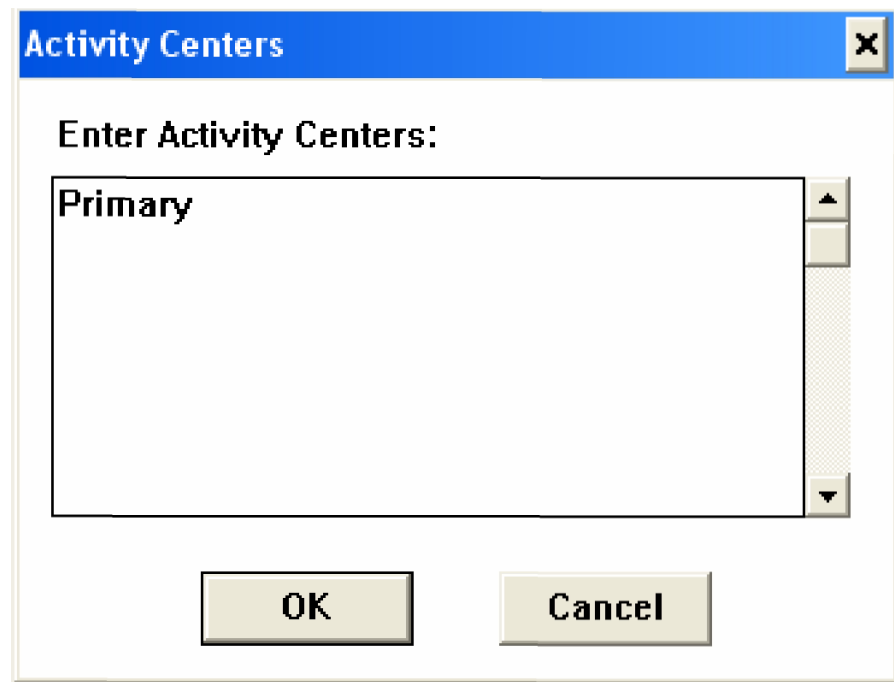


Рис. 3.4. Діалогове вікно «Activity Centers»

Заповнення глосарію можна робити або шляхом виділення кожної мітки і виклику для неї процедури Glossary Entry ... з меню Glossary, або за допомогою браузеру глосарію (Glossary => Glossary Browser ..., рис. 3.5).

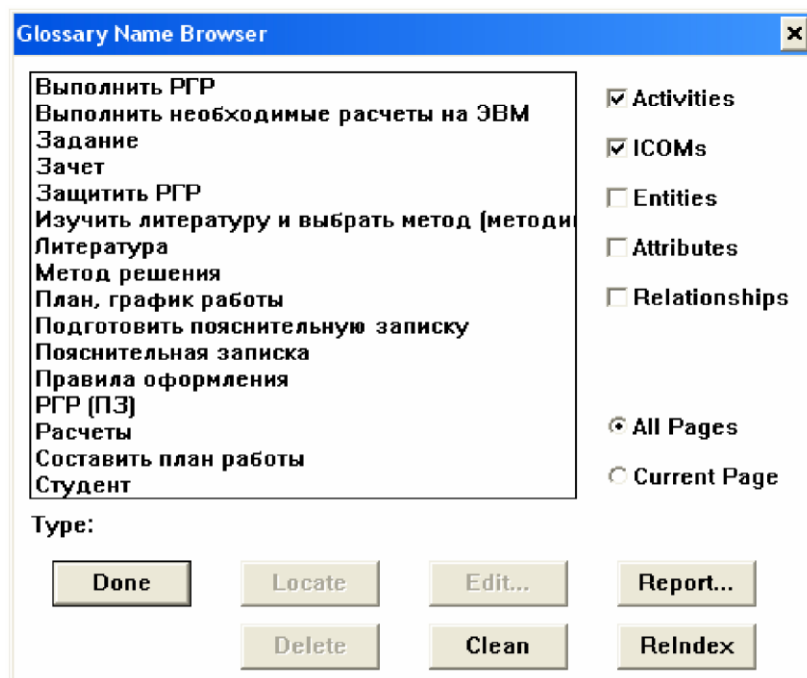


Рис. 3.5. Діалогове вікно «Glossary Name Browser»

При виклику браузера з'являється вікно зі списком імен, які використовуються в моделі. Є можливість вивести в цьому списку тільки назви блоків, сутностей, ярликів і т.д. За допомогою команди Edit кожне ім'я може бути доповнено визначенням, типом і кількісної характеристикою (рис. 3.6-3.7). Вихід з браузера відбувається при натисканні на кнопку Done.

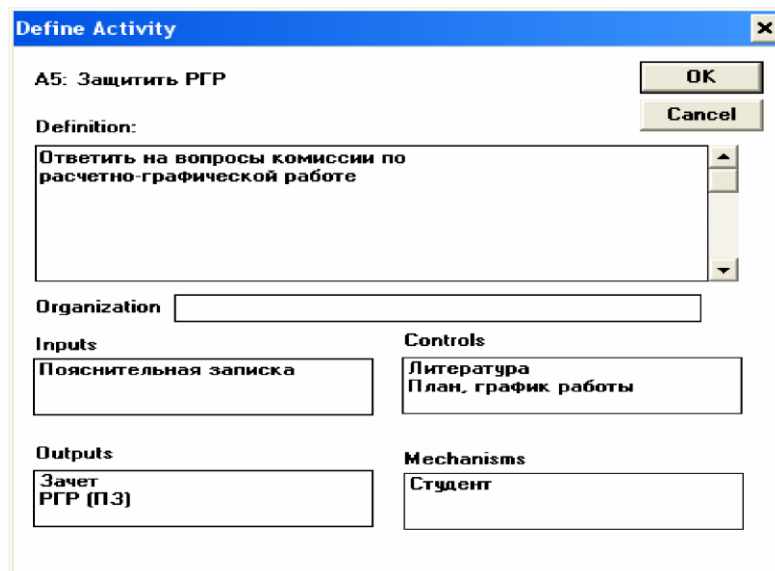


Рис. 3.6. Діалогове вікно «Define Activity»

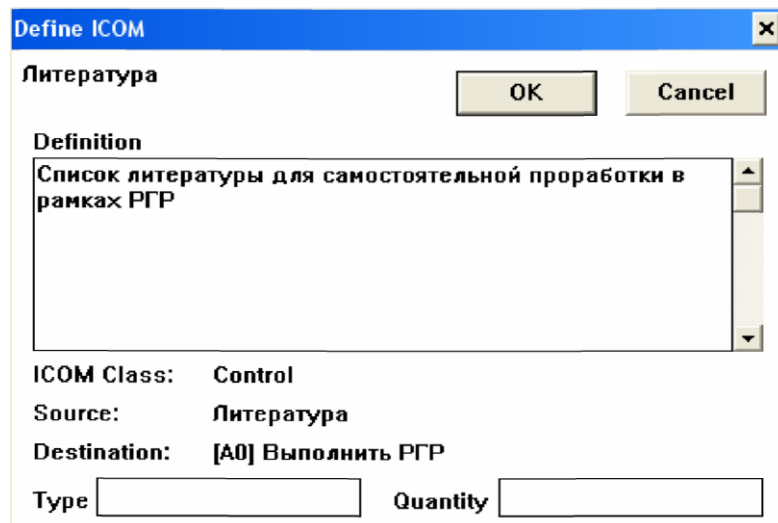


Рис. 3.7. Діалогове вікно «Define ICOM»

Для функціональних блоків необхідно визначити дані для функціонально-вартісного аналізу. Для цього є пункт Cost Information ... в меню Glossary. Сума витрат функціональних центрів розраховується автоматично і показується у вікні Total.

3.12. Перевірка синтаксису IDEF0-моделі

Команда *Validate* в пункті меню *File* переглядає всі сторінки в поточній моделі IDEF0 на правильність IDEF0 і SADT-синтаксису. Помилки синтаксису повідомляються згідно з форматом, який був обраний в діалозі *Validate IDEF0 Syntax* (рис. 3.8).

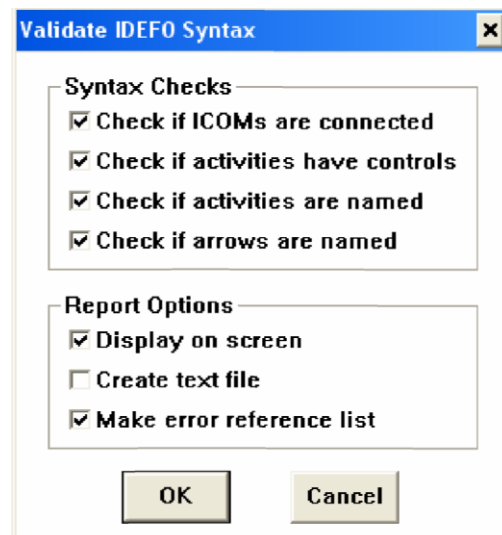


Рис. 3.8. Діалогове вікно «*Validate IDEF0 Syntax*»

Якщо вікно з результатами перевірки синтаксису (рис. 3.9) містить наведені нижче повідомлення, то це означає, що системний проект складений абсолютно правильно.

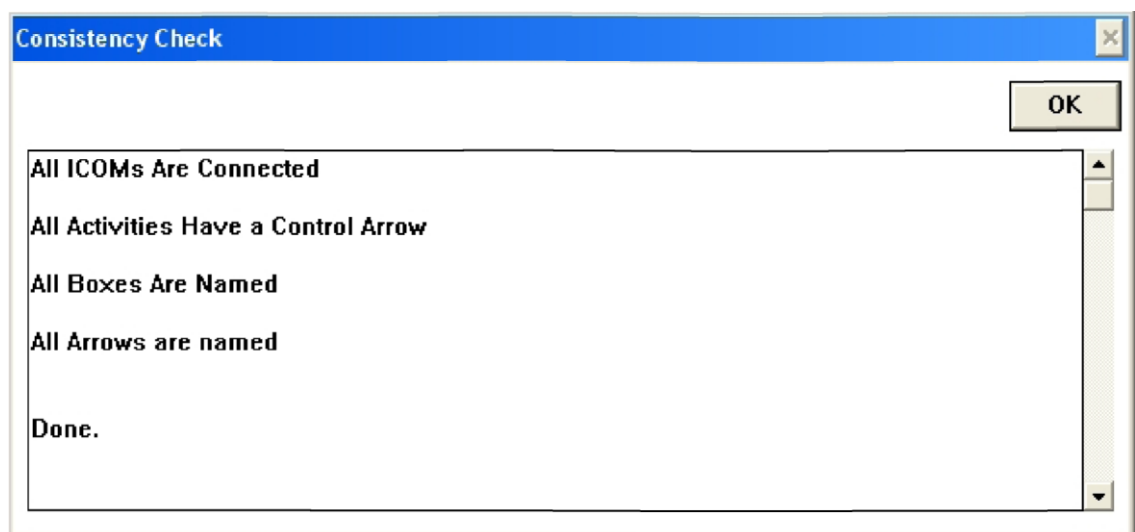


Рис. 3.9. Діалогове вікно «*Consistency Check*»

3.13. Збереження IDEF0-моделі

IDEFO-модель зберігається на диску за допомогою команд File => Save, File => SaveAs ... f, j File => Export ...

3.14. Приклад IDEF0-моделі «Виконати розрахунково-графічну роботу (РГР)»

На рис. 3.10 – 3.11 представлена IDEF0-модель «Виконати розрахунково-графічну роботу (РГР)».

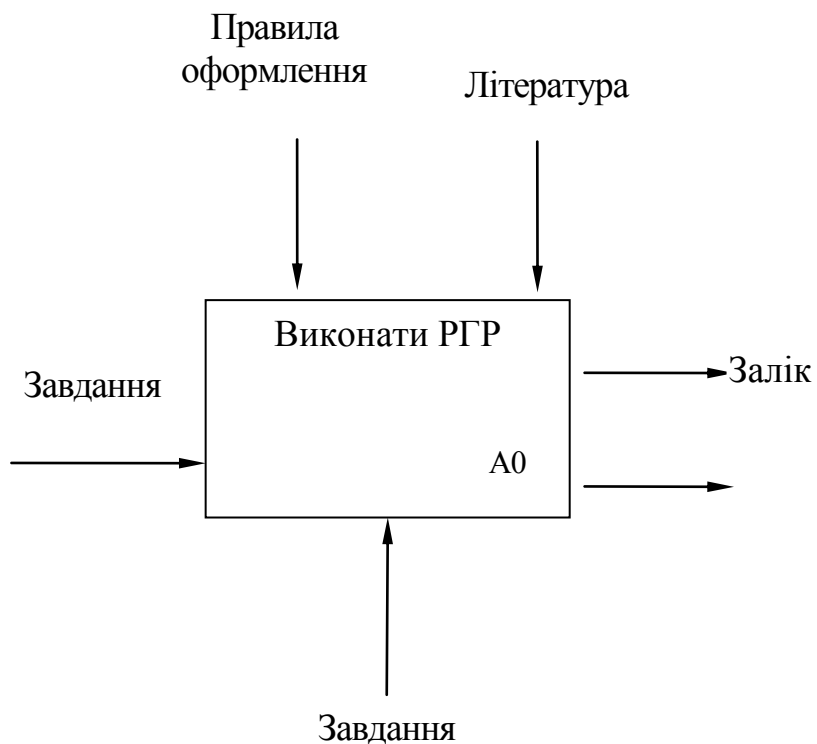


Рис. 3.10. Перший рівень функціональної моделі «Виконати РГР»

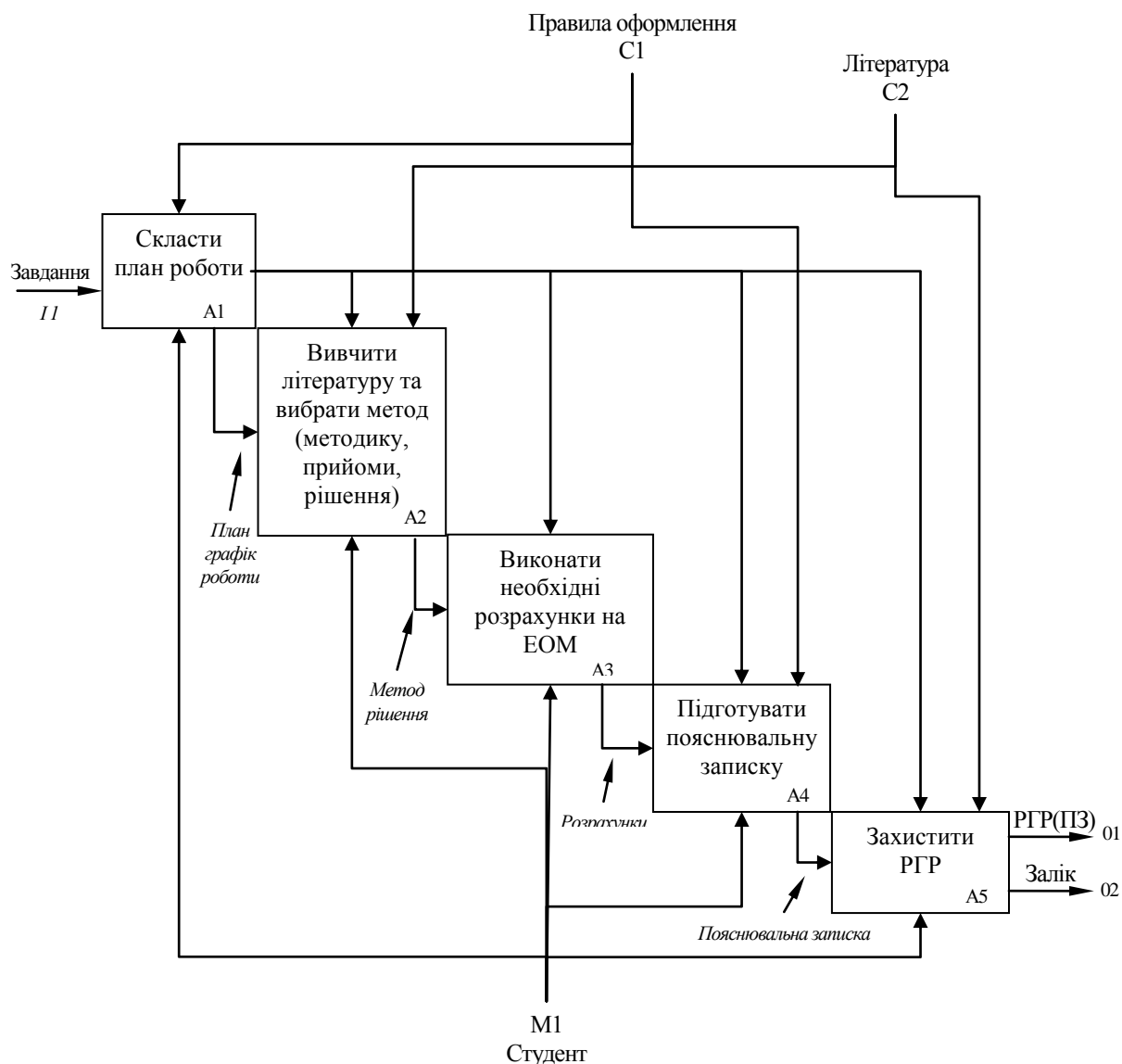


Рис. 3.11. Декомпозиція блоку АТ функціональної моделі «Виконати РГР»

Глосарій до функціональної моделі «Виконати РГР»

Об'єкти:

Завдання - постановка задачі на виконання РГР.

Залік - оцінка в балах, що виставляється студенту за РГР.

Література - список літератури для самостійного опрацювання в рамках РГР.

Метод рішення - метод вирішення поставленого завдання.

План, графік роботи - план виконання РГР.

Пояснювальна записка - оформлена відповідно до вимог пояснювальна записка до РГР.

Правила оформлення - методичні вказівки щодо виконання та оформлення РГР.

РГР (ПЗ) - розрахунково-графічна робота (пояснювальна записка).

Розрахунки - розрахунки, необхідні для вирішення поставленого завдання.

Студент - особа, що навчається у вищому навчальному закладі.

Дії:

АТ. Виконати РГР: виконати розрахунково-графічну роботу відповідно до завдання.

А1. Скласти план роботи: скласти план виконання розрахунково-графічної роботи.

А2. Вивчити літературу і вибрати метод (методику, прийоми рішення): виконати літературний огляд і вибрати метод вирішення поставленого завдання.

А3. Виконати необхідні розрахунки на ЕОМ: виконати розрахунки і побудувати необхідні графіки на ЕОМ.

А4. Підготувати пояснювальну записку: оформити пояснювальну записку до розрахунково-графічної роботи відповідно до встановлених вимог.

А5. Захистити РГР: відповісти на питання викладача по розрахунково-графічній роботі.

Ієрархічне представлення функціональної моделі «Виконати РГР»:

[A0 Виконати РГР.

[A1] Скласти план роботи.

[A2] Вивчити літературу і вибрати метод (методику, прийоми рішення).

[A3] Виконати необхідні розрахунки на ЕОМ [A4]. Підготувати пояснювальну записку [A5] Захистити РГР.

4. РОЗРОБКА ІНФОРМАЦІЙНОЇ МОДЕЛІ ПРЕДМЕТНОЇ ГАЛУЗІ В DESIGN/IDEF

4.1. Основні компоненти IDEF1X- моделі

Основними компонентами IDEF1X- моделі є:

1. Сутності, що представляють множини реальних або абстрактних предметів (людей, об'єктів, місць, подій, станів, ідей, пар предметів і т.д). Вони зображуються блоками. Виділяють два види сутностей: незалежні від ідентифікатора сутності; залежні від ідентифікатора сутності.

2. Відношення (зв'язки) між сутностями, зображуються лініями, що з'єднують блоки. Виділяють наступні види відношень: відношення, що ідентифікують зв'язки; відношення, які не ідентифікують зв'язки; відношення категоризації; неспецифічні відношення.

3. Атрибути - характеристики сутностей, що зображуються іменами всередині блоків. Атрибути можуть бути: неключовими; первинними ключами; альтернативними ключами; зовнішніми ключами.

4.1.1. Сутність

Сутність представляє множину реальних або абстрактних предметів (людей, об'єктів, місць, подій, станів, ідей, пар предметів і т.д.), що володіють загальними атрибутами або характеристиками.

Окремий елемент цієї множини називається екземпляром сутності.

Кожна сутність може мати будь-яку кількість відношень з іншими сутностями.

Сутність є незалежною, якщо кожен екземпляр сутності може бути однозначно ідентифікований без визначення його відношень з іншими сутностями.

Сутність називається залежною, якщо однозначна ідентифікація примірника сутності залежить від її відношення до іншої сутності.

Наприклад, сутності СПІВРОБІТНИК, ВІДДІЛ, ПОСАДА є незалежними, а сутність ШТАТНА-ОДИНИЦЯ - залежною (рис. 4.1).

Сутність володіє одним або декількома атрибутами, які або належать

сутності, або успадковуються через відношення.

Сутність володіє одним або декількома атрибутами, які однозначно ідентифікують кожен зразок сутності.

Кожна сутність може мати будь-якою кількістю відношень з іншими сутностями моделі.

Якщо зовнішній ключ цілком використовується в якості первинного ключа сутності або його частини, то сутність є залежною від ідентифікатора.

І навпаки, якщо використовується тільки частина зовнішнього ключа або взагалі не використовуються зовнішні ключі, то сутність є незалежною від ідентифікатора.

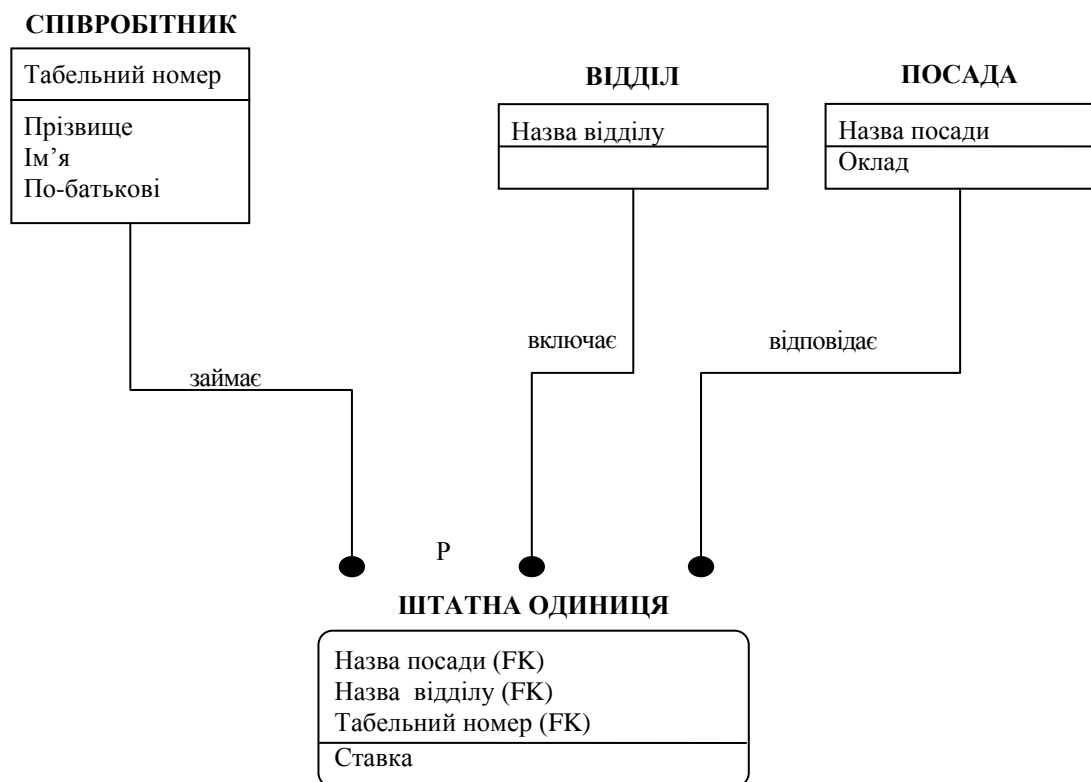


Рис. 4.1. Приклад фрагмента IDEF1X-моделі

4.1.2. Атрибути

Атрибут - це характеристика або елемент даних, що описує що-небудь в сутності. Атрибути можуть класифікуватися за належністю до одного з трьох різних типів: описові, що вказують, допоміжні. Описові атрибути представляють факти, внутрішньо властиві кожному екземпляру сутності. Вказуючі атрибути використовуються для присвоєння імені або позначення

екземплярів сутності. Допоміжні атрибути використовуються для зв'язку екземпляра однієї сутності з екземпляром іншої. Атрибути підкоряються суворо визначеними правилами.

Правила атрибутів:

- Кожен атрибут ідентифікується унікальним ім'ям. Одному і тому ж імені повинне відповідати одне і те ж значення. Одне і те ж значення не може відповідати різним іменам.

- Сутність може мати будь-яку кількість атрибутів. Кожен атрибут належить в точності одній сутності.

- Сутність може мати будь-яку кількість успадкованих атрибутів, але успадкований атрибут повинен бути частиною первинного ключа відповідної батьківської сутності або загальної сутності.

- Для кожного екземпляра сутності повинно існувати значення кожного його атрибута (правило неперетворення в нуль).

- Жоден з примірників сутності не може володіти більш ніж одним значенням для пов'язаного з нею атрибута (правило неповторення).

Сутність повинна мати атрибут або комбінацією атрибутів, чії значення однозначно визначають кожен екземпляр сутності. Ці атрибути утворюють *первинний ключ* сутності.

Наприклад, кожен з атрибутів *Табельний-номер*, *№-паспорта* і *№-поліса* може однозначно ідентифікувати примірник сутності *СЛУЖБОВЕЦЬ*, тому вони можуть бути атрибутами первинного ключа. Такі атрибути розміщуються нагорі списку і відділяються від інших атрибутів горизонтальною рисою. Інші атрибути сутності є *неключовими* (*Прізвище*, *Ім'я*, *По батькові*, *Дата-народження*).

При існуванні декількох можливих ключів один з них позначається як первинний ключ (атрибут *Табельний-номер* для сутності *службовець*), а решта - як альтернативні ключі (атрибут *№-паспорта* для сутності *службовець*). *Альтернативний ключ* - це ключ, який не є первинним ключем сутності. Кожному альтернативному ключу присвоюється унікальний цілий

номер. Цей ключ вказується за допомогою розміщення праворуч від кожного атрибута ключа укладених в дужки букв АК з номером альтернативного ключа. Атрибути зображуються у вигляді списку імен усередині блоку сутності. Атрибути, що визначають первинний ключ, розміщуються нагорі списку і відділяються від інших атрибутів горизонтальною рисою (рис. 4.2).



Рис. 4.2. Атрибути сутності СПІВРОБІТНИК

Атрибути, що з'являються в залежній сутності і є також первинним ключем іншої сутності (сутності-батька або загальної сутності) називаються *зовнішніми ключами*. Сутність може мати будь-яку кількість зовнішніх (успадкованих) атрибутів. У діаграмах моделі зовнішні ключі (Foreign Key) зображуються за допомогою розміщення всередину блоку сутності імен успадкованих атрибутів, після яких в дужках слідує буква FK. Якщо успадкований атрибут належить первинному ключу сутності-нащадка, то він розміщується вище горизонтальної лінії (а сутність зображується із закругленими кутами), а якщо немає, то - нижче (рис. 4.3).



Рис. 4.3. Атрибути сутності КОНТРАКТ

Процес переміщення первинного ключа батьківської або загальної

сутності в сутність-нащадок або сутність-категорію з відповідного відношення називається *міграцією ключа*.

Роль атрибута - це функція, яка виконується атрибутом в описі сутності, включаючи функцію мігративності, приналежності первинного ключа, альтернативного ключа. Зазвичай ім'я ролі атрибута призначається для кожного випадку, коли окремий атрибут успадковується більше одного разу (тобто коли сутність-нащадок має кілька різних відношень з однією і тією ж сутністю-батьком).

Наприклад, сутність СПІВРОБІТНИК в якості сутності-батька має подвійну відношенням з сутністю РОБОЧА ГРУПА. Один і той же співробітник може бути як керівником, так і виконавцем робочої групи. Імена ролей *Керівник* і *Виконавець* можуть призначатися для розрізнення двох успадкованих атрибутів *Табельний-номер* (рис. 4.4). Імена ролей можуть використовуватися і при одній появі успадкованого атрибута для більш точного вираження його сенсу в контексті сутності-нащадка, але це не є обов'язковим.

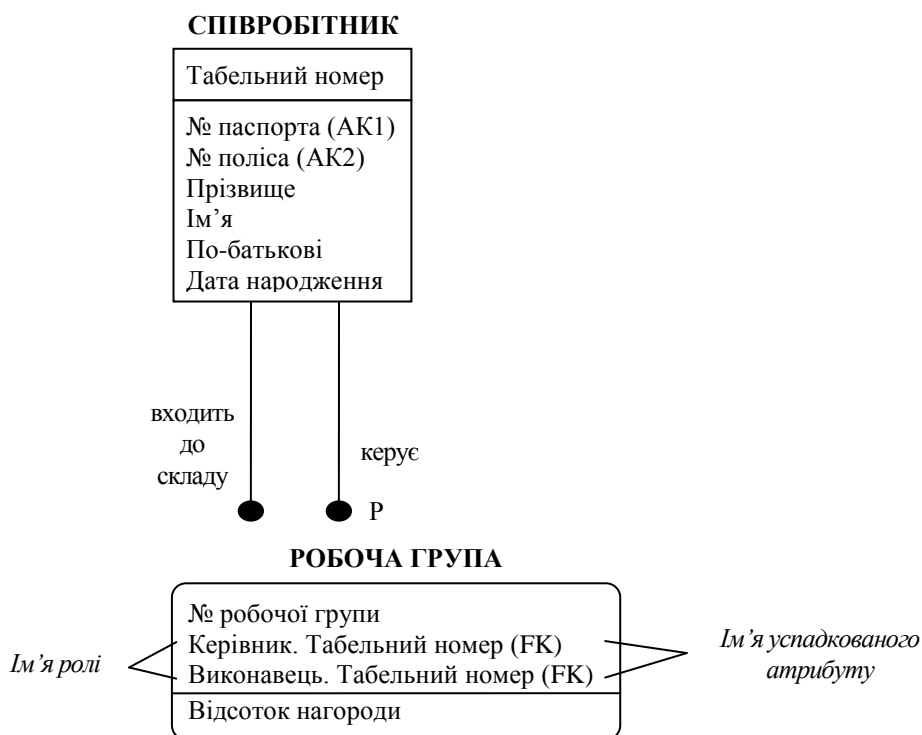


Рис. 4.4. Фрагмент IDEF1X-моделі

Правила ключів:

- кожна сутність повинна мати первинний ключ;
- кожна сутність може мати будь-яке число альтернативних ключів;
- первинний і альтернативний ключ може складатися з одного атрибута або комбінації атрибутів;
 - окремий атрибут може бути частиною більш ніж одного ключа, первинного або альтернативного;
 - атрибути, що входять у первинні або альтернативні ключі сутності, можуть бути власними для сутності або успадковуватися через відношення (зовнішніми);
 - первинні або альтернативні ключі повинні містити тільки необхідні для однозначної ідентифікації атрибути, тобто при виключенні з ключа будь-якого атрибута не всі екземпляри сутності можуть бути однозначно визначені (правило найменшого ключа);
 - якщо первинний ключ складається більш ніж з одного атрибута, то значення будь-якого неключевого атрибута повинно функціонально залежати від усього первинного ключа, тобто якщо первинний ключ відомий, то відомо значення кожного неключевого атрибута, і значення неключевого атрибута не може бути визначено за допомогою тільки частини первинного ключа (правило повної функціональної залежності);
 - кожен неключовий атрибут повинен функціонально залежати тільки від первинного і альтернативних ключів, тобто значення неключевого атрибута не може визначатися значенням іншого неключевого атрибута (правило відсутності транзитивної залежності).

4.1.3. Відношення (зв'язок)

Відношення зв'язку, що називається також відношенням батько-нащадок, - це зв'язок між сутностями, при якій кожен екземпляр сутності-батька асоційований з довільною (в тому числі нульовою) кількістю примірників сутності-нащадка, а - кожен екземпляр сутності-нащадка асоційований в точності з одним екземпляром сутності-батька.

Якщо екземпляр сутності-нащадка однозначно визначається своїм зв'язком з сутністю-батьком, то зв'язок називається ідентифікуючим, а в іншому випадку - неідентифікуючим.

Зв'язок зображується лінією, що проводиться між сутністю-батьком і сутністю-нащадком з крапкою на кінці лінії у сутності-нащадка.

Відношенню дається ім'я, яке виражається граматичним оборотом дієслова. Ім'я відношення завжди формується з точки зору батьків, так що може бути утворено пропозицію, якщо з'єднати ім'я сутності-батька, ім'я відношення, вираз потужності і ім'я сутності-нащадка.

Потужність визначає скільки примірників сутності-нащадка може існувати для кожного екземпляра сутності-батька.

У IDEF1X можуть бути виражені наступні потужності відношення (таблиця 2.6):

- кожен екземпляр сутності-батька може мати нуль, один або більше пов'язаних з ним екземплярів сутності-нащадка (ця потужність встановлюється за замовчуванням);

- кожен екземпляр сутності-батька повинен мати не менше одного пов'язаного з ним екземпляра сутності-нащадка (буква P (Positive), розміщена поруч з крапкою);

- кожен екземпляр сутності-батька може мати не більше одного пов'язаного з ним екземпляра сутності-нащадка (буква Z (zero), вміщена поруч з крапкою);

- кожен екземпляр сутності-батька пов'язаний з деяким фіксованим числом екземплярів сутності-нащадка. Якщо потужність в точності дорівнює деякому числу N, це число (ціле, позитивне) поміщається близько крапки.

Ідентифікуючий зв'язок між сутністю-батьком і сутністю-нащадком зображується суцільною лінією (рис. 4.5).

Сутність-нащадок в ідентифікуючому зв'язку є залежною від ідентифікатора сутністю. Сутність-батько в ідентифікуючому зв'язку може бути як незалежною, так і залежною від ідентифікатора сутністю (це

визначається її зв'язками з іншими сутностями).

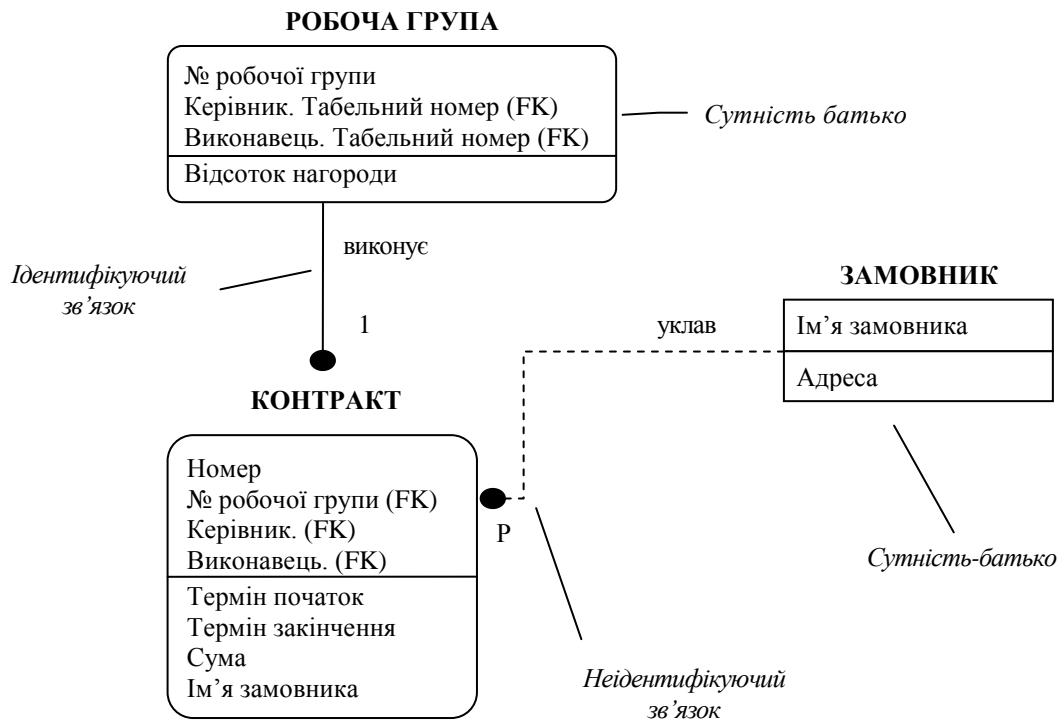


Рис. 4.5. Фрагмент IDEF1X-моделі

Пунктирна лінія зображує неідентифікуючий зв'язок (рис. 4.5). Сутність-нащадок в неідентифікуючому зв'язку буде незалежною від ідентифікатора, якщо вона не є також сутністю-нащадком в будь-якому ідентифікуючому зв'язку.

Відношення повної категоризації - це відношення між двома або більше сутностями, в якому кожен екземпляр однієї сутності, яка називається загальною сутністю, пов'язаний в точності з одним екземпляром однієї і тільки однієї з інших сутностей, що називаються сутностями-категоріями (рис. 4.6).

Кожен екземпляр загальної сутності і пов'язаний з ним екземпляр однієї з категорних сутностей відображують один і той же предмет реального світу і тому володіють одним і тим же унікальним ідентифікатором.

Для сутності-категорії потужність не вказується, оскільки вона завжди дорівнює нулю або одиниці. Наприклад, **ЗАМОВНИК** є спільною сутністю, а **ЗАРУБІЖНЕ-ПІДПРИЄМСТВО** і **ВІТЧИЗНЯНЕ-ПІДПРИЄМСТВО** є

сутностями-категоріями. Значення деякого атрибута в екземплярі загальної сутності визначає, з яким з можливих сутностей-категорій він пов'язаний. Цей атрибут називається дискримінатором відношення категоризації. В даному прикладі дискримінатором є *Національна-належність*. Якщо існує екземпляр загальної сутності, не пов'язаний ні з яким примірником з сутностей-категорій, то таке відношення називається відношенням неповної категоризації. На неповноту множин категорій на діаграмі вказує коло, підкреслене один раз.



Рис. 4.6. Відношення повної категоризації в нотації IDEF1X

Неспецифічне відношення, що називається також відношенням багато до багатьох, - це зв'язок між двома сутностями, при якій кожен екземпляр першої сутності пов'язаний з довільною (в тому числі нульовою) кількістю примірників іншої сутності, а кожен примірник іншої сутності пов'язаний з довільною (в тому числі нульовою) кількістю примірників першої сутності. Наприклад, якщо службовець може бути зайнятий в багатьох проектах, а в проекті може бути зайнято багато службовців, то відношення між сутностями **СЛУЖБОВЕЦЬ** і **ПРОЕКТ** є неспецифічним відношенням. На більш пізніх стадіях побудови моделі всі неспецифічні відношення повинні бути деталізовані в специфічні.

4.2. Розробка концептуальної схеми бази даних в Design/IDEF

4.2.1. Запуск IDEF1X моделювання

Перший крок у використанні Design/IDEF для IDEF1X моделювання - відкриття нової IDEF1X-сторінки у новій або існуючій IDEF1X-моделі.

4.2.2. Створення нової IDEF1X-моделі

Щоб почати створення нової IDEF1X-моделі:

- а) запустить *Design/IDEF*;
- б) Виберіть *New* в меню *File*;
- в) в діалоговому вікні «*Select New Page Type*» (рис. 4.7) для створення інформаційної моделі в списку *Methodology* виберіть IDEF1X і підтвердіть вибір клавішею *OK*;
- г) з'являється чиста IDEF1X-сторінка для створення інформаційної моделі.

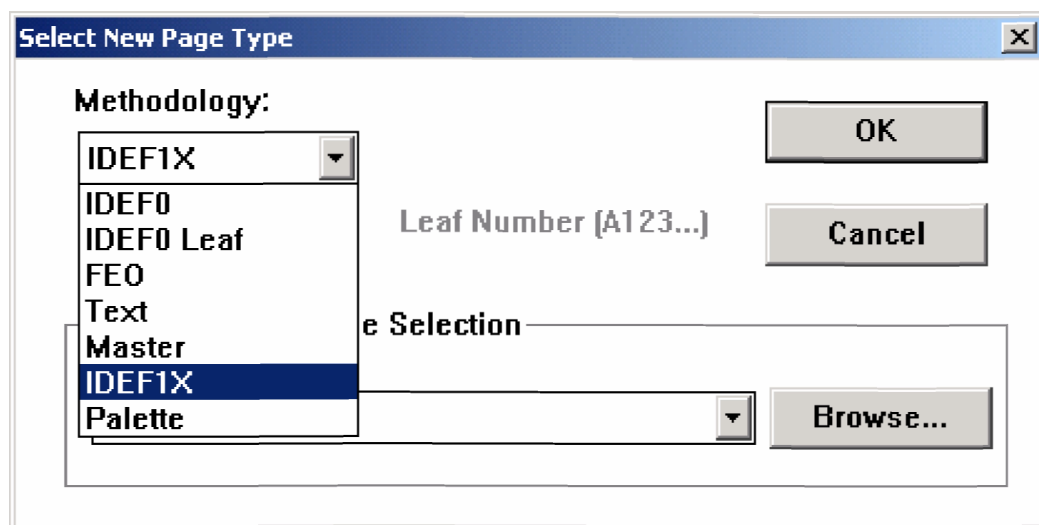


Рис. 4.7. Діалогове вікно «*Select New Page Type*»

Створення об'єктів IDEF1X-класу відбувається шляхом виклику пункту *Entity* меню *Create* або натисканням відповідної кнопки на панелі інструментів (рис. 4.8).

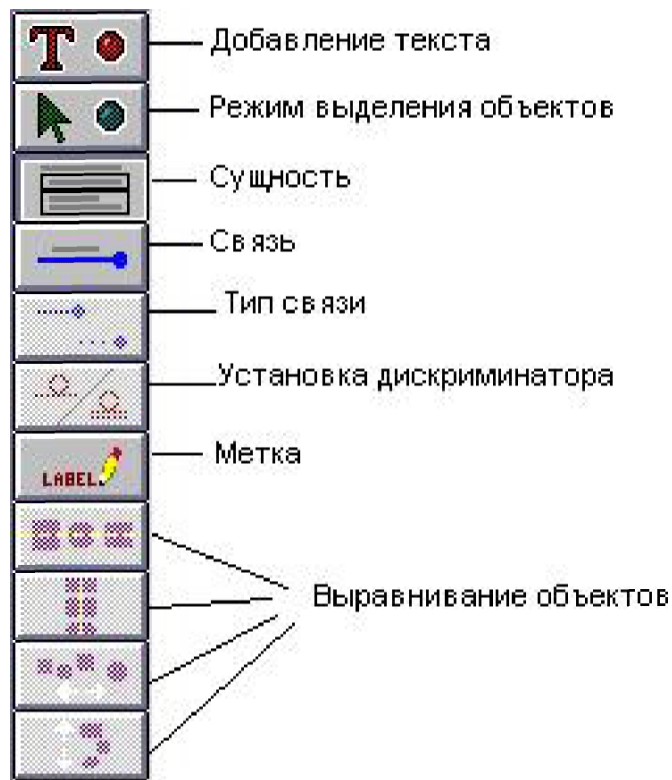



Рис. 4.8. Інструменти для створення IDEF1X-моделі

4.2.3. Створення об'єктів IDEF1X класу

Створення об'єктів-сутностей. Щоб створити нову сутність в IDEF1X-моделі:

а) Виберіть *Entity* в меню *Create* або натисніть кнопку  на функціональній панелі інструментів;

б) позначте за допомогою миші прямокутний блок на сторінці діаграми;

в) заповніть поля у вікні «*Define Entity*» (Рис. 4.9): *Name* - це унікальне ім'я, за допомогою якого сутність буде розпізнаватися в моделі; *Aliases* - псевдонім під яким сутність може бути відома; *Definition* - визначення сутності, яке зазвичай використовується в предметній галузі; *Attributes* - список атрибутів даної сутності. Щоб ввести список атрибутів потрібно натиснути на кнопку *Add*.

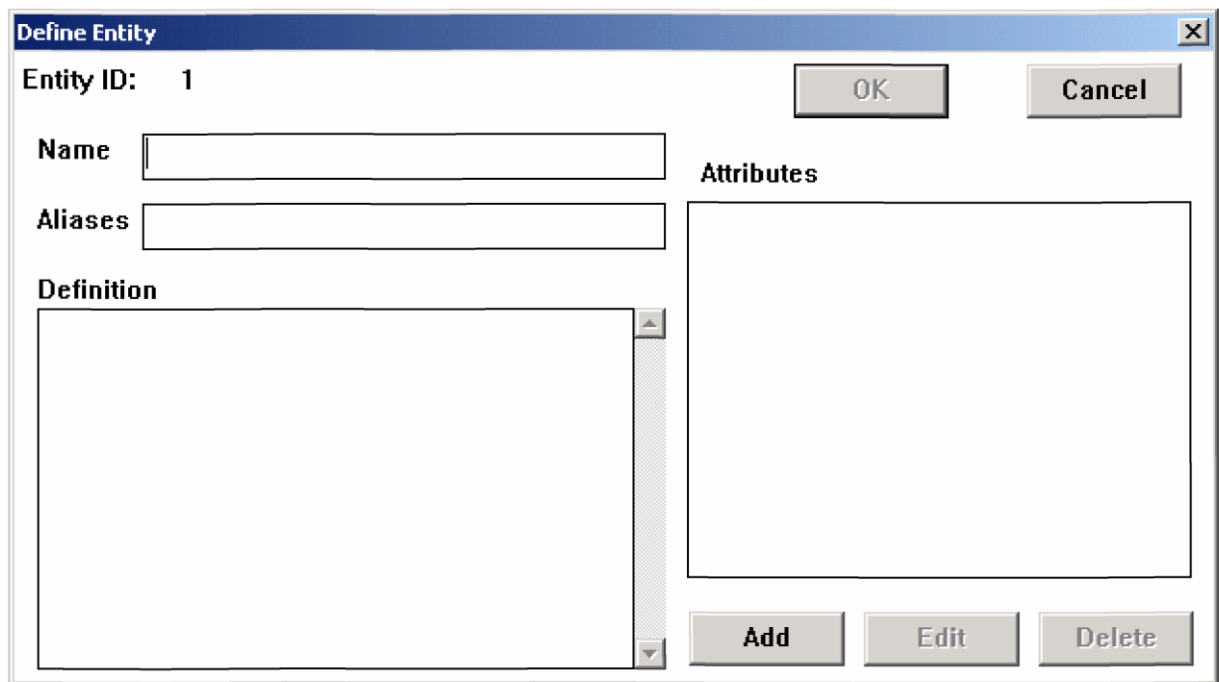


Рис. 4.9. Діалогове вікно «Define Entity»

Створення атрибутів. Створення атрибутів виконується у вікні «Define Attribute» (рис. 4.10).

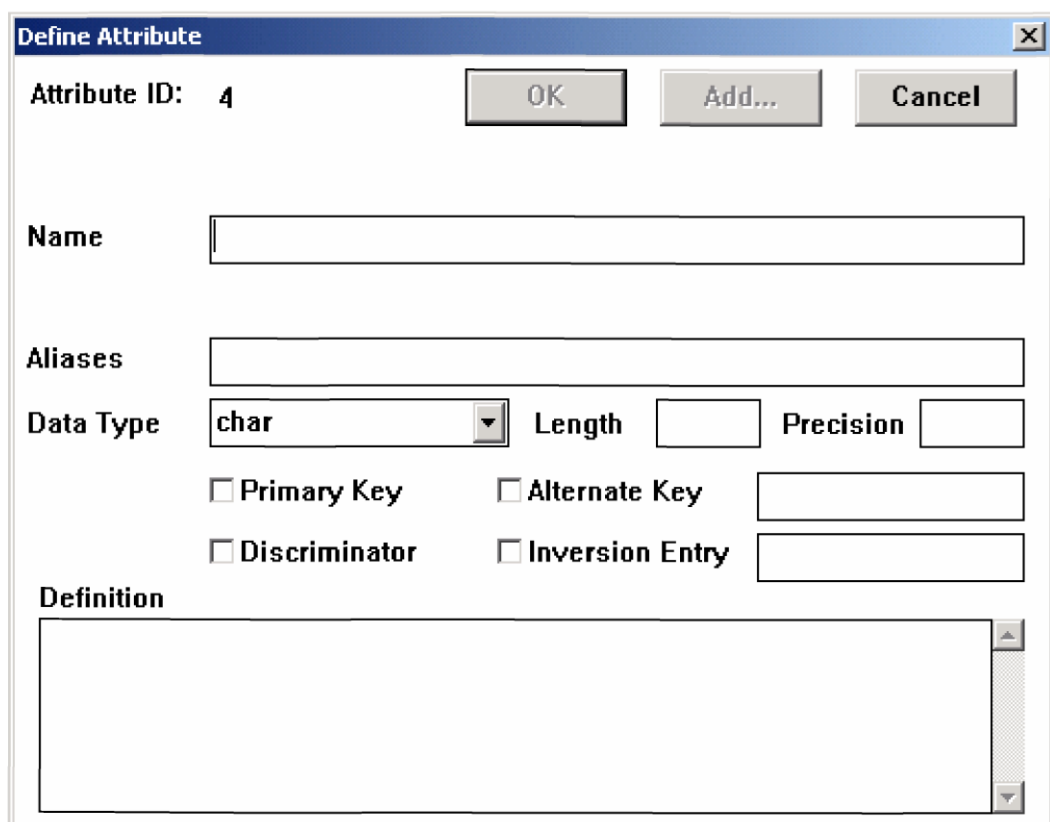


Рис. 4.10. Діалогове вікно «Define Attribute»

Вікно «Define Attribute» містить: Name - ім'я атрибута; Aliases - псевдонім атрибута; Datatype - тип значення атрибута; Length - довжина значення атрибута; Precision - кількість чисел після крапки, для атрибутів числового типу; Primary Key - первинний ключ; Alternate Key - альтернативний ключ; Definition - визначення атрибута, яке зазвичай використовується в предметній галузі; Discriminator - дискримінатор.

Наприклад, типи даних (Datatype), які використовуються в мові SQL для задавання типу значення атрибута, досить різноманітні:

1. INT - ціле число довжиною 4 байта, яке надається при виведенні максимально Length цифрами.

2. SMALLINT - ціле число довжиною 2 байта, яке надається при виведенні максимально Length цифрами.

3. FLOAT - дійсне число, яке надається при виведенні максимально Length символами з Precision цифрами після десяткової крапки.

4. CHAR - рядок символів фіксованої довжини розміром Length символів.

5. VARCHAR - рядок символів змінної довжини максимальним розміром до Length символів.

6. BLOB (Binary Large Object) - масив довільних (двійкових) байтів (максимальний розмір залежить від реалізації, зазвичай це 65535 байт); цей тип даних може використовуватися, наприклад, для зберігання зображень.

7. DATE - астрономічна дата.

8. TIME - астрономічний час.

Після задавання атрибуту його властивостей необхідно натиснути клавішу OK для збереження заданих властивостей. У вікні «Attributes» визначається сутність з'являються написи, які вказують на наявність атрибутів і їх властивостей.

Створення відношень (зв'язків). Щоб створити нове відношення (зв'язок) в IDEF1X моделі:

a) Виберіть *Relationship* в меню Create або натисніть кнопку  на

функціональній панелі інструментів;

б) з'єднайте курсором миші сутності, між якими встановлюється відношення (зв'язок);


в) заповніть поля у вікні «*Define Relationship*»(рис. 4.11): Relationship - ім'я відношення; Inverse - ім'я сутності-нащадка, з яким встановлюється зв'язок відношення (вказується в неспецифічних відношеннях); Definition - визначення зв'язку, яке зазвичай використовується в предметній галузі; Relationship Type - тип зв'язку (ідентифікуючий, неідентифікуючий або неспецифічний); Relationship Cardinality - потужність встановлюваного зв'язку.

The image shows a dialog box titled "Define Relationship". It contains the following elements:

- Entity:** СУЩНОСТЬ_1
- Relationship:** [Empty text box]
- Entity:** СУЩНОСТЬ_2
- Inverse:** [Empty text box]
- Separate Lines**
- Definition:** [Large empty text area]
- Relationship Type:**
 - Identifying
 - Non-Identifying
 - Null Allowed
 - Non-Specific
- Relationship Cardinality:**
 - Zero, One, or Many
 - Zero or One [Z]
 - One or Many [P]
 - Exactly [Empty box]
- Buttons:** OK, Cancel

Рис. 4.11. Діалогове вікно «*Define Relationship*»

Після створення зв'язків при наявності атрибутів в інформаційних блоках відбувається обмін атрибутами, тобто головний ключ блоку з якого виходить відношення перетворюється у зовнішній ключ (Foreign key) у блоці куди входить відношення. Поміняти тип відношення (з ідентифікуючого на

неідентифікуючий) можна за допомогою меню Create або кнопкою  на панелі інструментів.

4.3. Збереження IDEF1X-моделі

IDEF1X-модель зберігається на диску за допомогою команд File => Save, File => Save As ... або File => Export ...

4.4. Генерація sql-скрипта на IDEF1X-моделі

SQL-скрипт: - файл, який містить послідовність команд на мові SQL для створення бази даних, представлена у вигляді IDEF1X-моделі

Для генерації *SQL*-скрипта для створення бази даних на IDEF1X-моделі виберіть пункт меню File => Export. У діалоговому вікні «File Export» (рис. 4.12) в списку Format виберіть SOL Files (*.SQL), введіть ім'я *SQL*-скрипта і підтвердіть вибір клавішею OK.

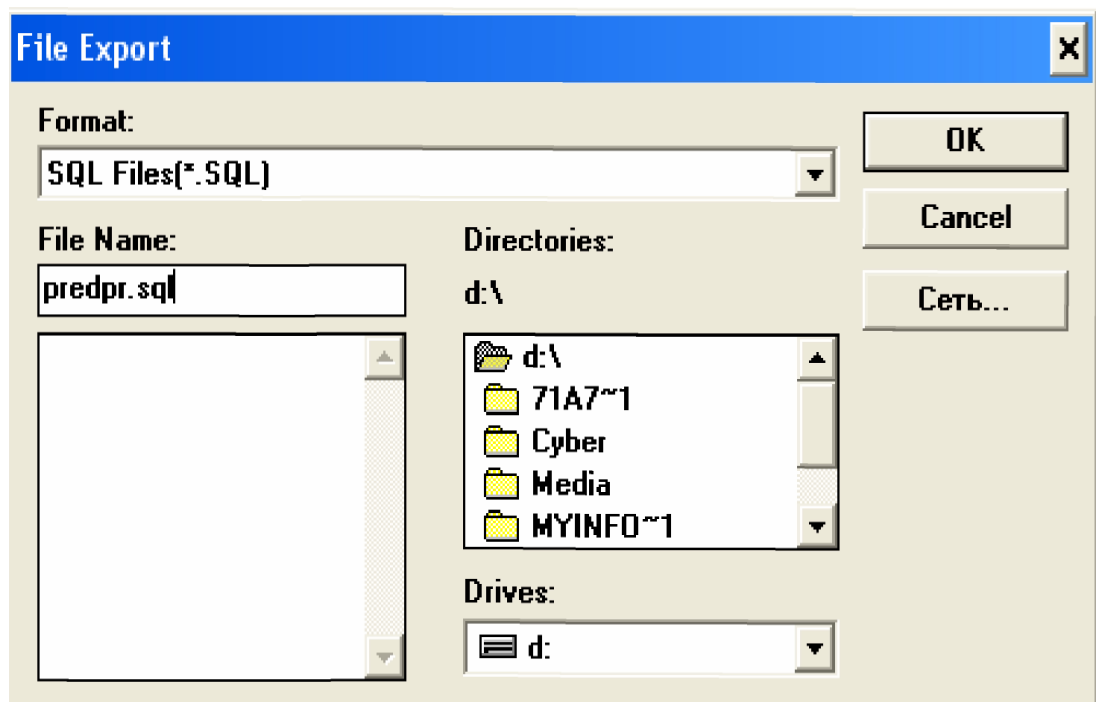


Рис. 4.12. Діалогове вікно «Select New Page Type»

4.5. Приклад IDEF1X-моделі

Приклад концептуальної схеми бази даних в нотації IDEF1X наведено на рис. 4.13.

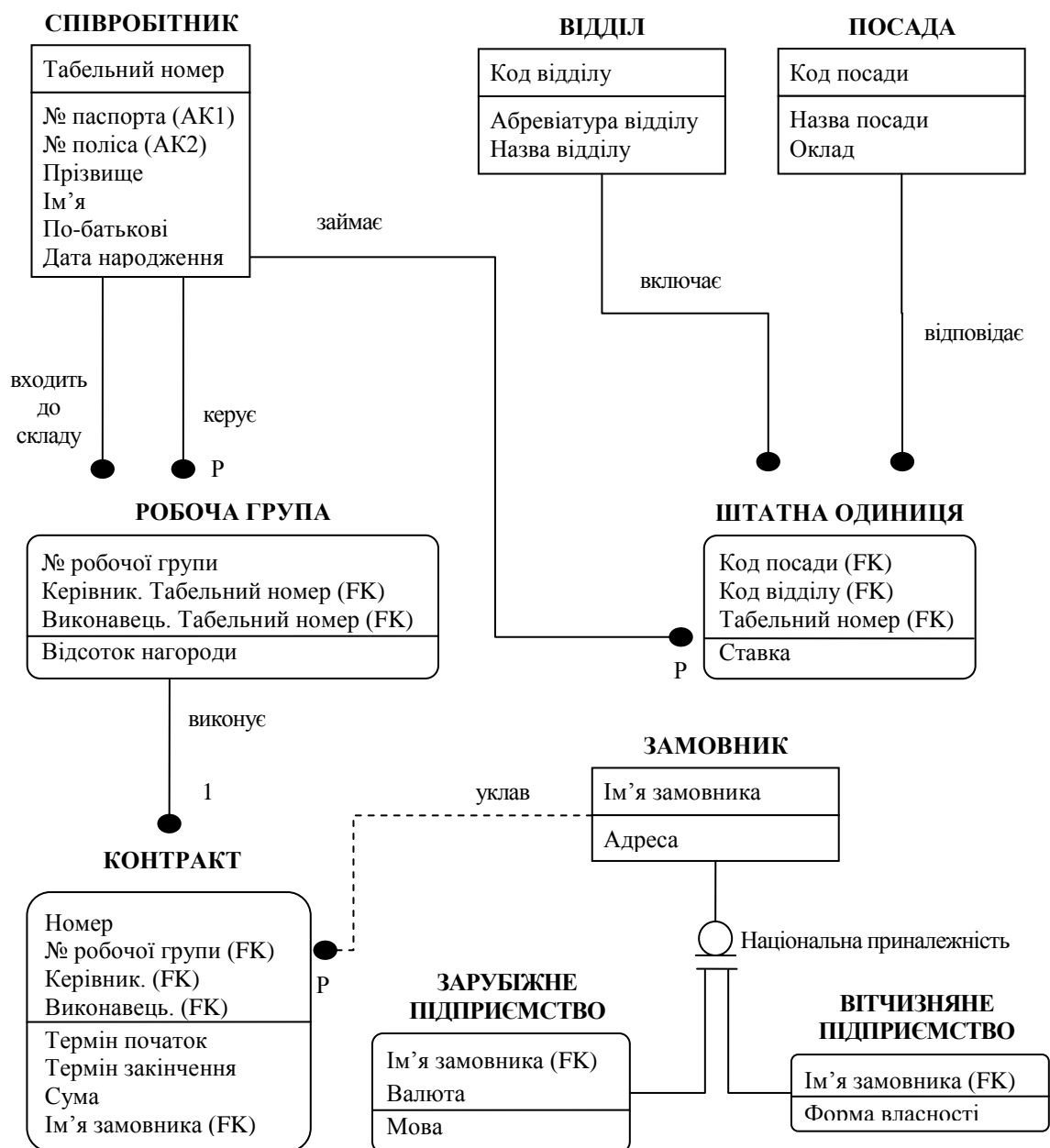


Рис. 4.13. Приклад концептуальної схеми бази даних в нотації IDEF1X

5. ЛАБОРАТОРНИЙ ПРАКТИКУМ

5.1. Лабораторна робота №1: Методика побудови функціональної моделі предметної галузі для проектування автоматизованої системи управління

Мета роботи: вивчити принципи розробки та формалізації предметної галузі у вигляді функціональної моделі в нотації IDEF0; освоїти прийоми побудови функціональної моделі предметної галузі.

5.1.1. Послідовність виконання для лабораторної роботи

1. Ознайомитися із запропонованим варіантом опису предметної галузі. Проаналізувати предметну галузь, уточнити і доповнити її, керуючись власним досвідом, консультаціями та іншими джерелами.

2. Виконати структурне розбиття предметної галузі на окремі підрозділи (відділи, служби, підсистеми, групи та ін.), відповідно до виконуваних ними функцій.

3. Визначити завдання і функції системи в цілому і функції кожного підрозділу (підсистеми).

4. Виконати словесний опис роботи кожного підрозділу (підсистеми), алгоритмів і сценаріїв виконання ними окремих робіт.

5. Побудувати діаграми робіт і діаграми потоків даних в нотації *Йордона - де Марко* і IDEF0 для всієї інформаційної системи в цілому і для окремих сценаріїв робіт, що відображають логіку і взаємовідношення підрозділів (підсистем).

6. Вивчити методику побудови інформаційної моделі предметної галузі за допомогою пакета *Design/IDEF*.

7. Реалізувати модель за допомогою пакета *Design/IDEF*, попередньо встановивши атрибути сторінки IDEF0-моделі, скориставшись пунктом меню Edit => Set Page Attributes. Встановіть значення наступних атрибутів сторінки: name - найменування предметної галузі (варіанта

завдання); user - виконав (П.І.Б. студента); project - найменування лабораторної роботи; status - Publication; revision - перевірів (П.І.Б викладача).

8. Заповнити глосарій *IDEF0*-моделі і перевірити її синтаксис.

9. Експортувати *IDEF0*-модель у формат Windows MetaFile (*.wmf) скориставшись пунктом меню File => Export.

10. Сформувати *IDEF0* звіт і Glossary Report, скориставшись пунктом меню File => Report.

11. Оформити звіт по лабораторній роботі.

5.1.2. Вимоги до звіту

Звіт з лабораторної роботи повинен містити:

1. Титульну сторінку.
2. Мету роботи.
3. Варіант завдання.
4. Склад підрозділів (підсистем) інформаційної системи;
5. Перелік функцій і завдань системи в цілому і кожного підрозділу (підсистеми) окремо; докладний опис роботи кожного підрозділу (підсистеми), відношення їх між собою, опис окремих сценаріїв робіт.

6. Діаграми робіт і діаграми потоків даних в нотації *Йордона - де Марко* і *IDEF0* для всієї інформаційної системи в цілому і для окремих сценаріїв робіт, що відображають логіку і взаємовідношення підрозділів (підсистем).

7. *IDEF0*-модель предметної галузі в форматі Windows MetaFile (*.wmf).

8. *IDEF0* звіт і Glossary Report.

5.1.3. Контрольні питання

1. Наведіть завдання методології структурного аналізу даних?
2. Наведіть види зв'язків в методології *IDEF0*?

3. Наведіть призначення методології діаграм потоків даних?
4. Що таке потік даних в методології DFD?
5. Яка функція сховища даних в DFD?
6. У чому подібність і відмінність методології структурного аналізу даних і діаграм потоків даних?
7. Яке призначення має функціональна модель в процесі розробки ПЗ?
8. Перерахуйте основні компоненти функціональної моделі.
9. Опишіть правила формування функціональних блоків (ієрархія, нумерація, позначення).
10. Опишіть правила створення дуг (напрямок, тип інтерфейсу, позначення).

5.2. Лабораторна робота №2: Методика побудови інформаційної моделі предметної галузі для проектування автоматизованої інформаційної системи

Мета роботи: вивчити принципи розробки та формалізації предметної галузі у вигляді інформаційної моделі IDEF1X для побудови АІС; освоїти прийоми побудови інформаційної моделі предметної галузі.

5.2.1. Послідовність виконання лабораторної роботи

1. Згідно варіанту завдання, виділити необхідний набір сутностей, що відображають предметну галузь та інформаційні потреби користувачів.
2. Визначити необхідний набір атрибутів кожної сутності, виділивши ідентифікують атрибути.
3. Класифікувати атрибути кожної сутності (описові, що вказують, допоміжні).
4. Визначити сутність виду підтип /супертип, де це необхідно.
5. Визначити зв'язки між сутностями.
6. Проаналізувавши структуру зв'язків, виключити надлишкові.
7. Визначити множинність і умовність зв'язків.

8. Дати формулювання зв'язків з точки зору кожної сутності.
9. Формалізувати зв'язку виду $1:1$, $1:M$, $M:N$.
10. Побудувати ER-діаграми моделі бази даних: в нотації Чена; в нотації Мартіна; в нотації *IDEF1X*, в нотації Баркера.
11. Описати модель бази даних на мові інфологічного моделювання.
12. Вивчити методику побудови інформаційної моделі предметної галузі за допомогою пакета *Design/IDEF*.
13. Реалізувати *ER*-діаграму моделі бази даних в нотації *IDEF1X* за допомогою пакета *Design/IDEF*, попередньо встановивши атрибути сторінки *IDEF1X*-моделі, скориставшись пунктом меню Edit => Set Page Attributes. Встановіть значення наступних атрибутів сторінки: name - найменування предметної галузі (варіанта завдання); user - виконав (П.І.Б. студента); project - найменування лабораторної роботи; status - Publication; revision - перевірів (П.І.Б викладача).
14. Експортувати *IDEF1X*-модель в формат *WindowsMetaFile (*.wmf)* і згенерувати *sql*-скрипт для створення бази даних, скориставшись Пунктом меню File => Export.
15. Сформувати *IDEF1X* звіт, скориставшись пунктом меню *File => Report*.
16. Оформити звіт по лабораторній роботі.

5.2.2. Вимоги до звіту

Звіт з лабораторної роботи повинен містити:

1. Титульну сторінку.
2. Мету роботи.
3. Варіант завдання.
4. *ER*-діаграми моделі бази даних: в нотації Чена; в нотації Мартіна; в нотації *IDEF1X*, В нотації Баркера.
5. Опис моделі бази даних на мові інфологічного моделювання.
6. *IDEF1X*-модель бази даних в форматі *Windows MetaFile (*.wmf)*.

7. IDEF1X звіт.

5.2.3. Контрольні питання

1. Які завдання, які вирішуються на етапі концептуального проектування?
2. Дайте характеристику моделі типу «сутність - зв'язок».
3. Опишіть відомі форми представлення даних.
4. Яке призначення має інформаційна (концептуальна) модель в процесі проектування автоматизованої інформаційної системи?
5. Перерахуйте основні складові інформаційної (концептуальної) моделі.
6. У чому полягає відмінність поняття типу сутності і елемента сутності?
7. Наведіть способи подання сутності?
8. Опишіть правила формування сутностей (типи, елементи, нумерація, позначення).
9. Які правила атрибутів?
10. Як класифікуються атрибути?
11. Які фундаментальні види зв'язків?
12. Як формалізується зв'язок 1:1?
13. Як формалізується зв'язок 1:M?
14. Як формалізується зв'язок M:N?
15. Опишіть правила створення відношень (напрямок, тип, позначення).
16. Що таке підтип і супертип?
17. Поясніть принцип роботи і порядок створення дискримінатора.

5.3. Варіанти завдань до лабораторного практикуму

Кожен варіант завдання представлений у вигляді опису предметної галузі та переліку запитів, які повинні виконуватися в інформаційній системі.

5.3.1. Інформаційна система Вузу

Опис предметної галузі. Студенти, організовані в групи, навчаються на одному з факультетів, який очолює деканат, у функції якого входить контроль за навчальним процесом. У навчальному процесі беруть участь викладачі кафедр, які адміністративно відносяться до одного з факультетів університету. Викладачі поділяються на такі категорії: асистенти, викладачі, старші викладачі, доценти, професори. Асистенти і викладачі можуть навчатися в аспірантурі, старші викладачі, доценти, можуть очолювати наукові теми, професори - наукові напрямки. Викладачі будь-якої категорії свого часу могли захистити кандидатську, а доценти та професори і докторську дисертацію, при цьому викладачі можуть займати посади доцента і професора тільки у випадку, якщо вони мають відповідно звання доцента і професора.

Навчальний процес регламентується навчальним планом, у якому вказується, які навчальні дисципліни на яких курсах і у яких семестрах читаються для студентів кожного року набору, із визначенням кількості годин на кожен вид занять з дисципліни (види занять: лекції, семінари, лабораторні роботи, консультації, курсові роботи, ІР і т.ін.) і форми контролю (залік, іспит). Перед початком навчального семестру деканати роздають на кафедри навчальні доручення, у яких вказуються кафедри (не обов'язково пов'язані з даним факультетом), які дисципліни і для яких груп повинні проводити в черговому семестрі. Керуючись ними, на кафедрах здійснюється розподіл навантаження, при цьому з однієї дисципліни в одній групі різні види занять можуть проводити один або кілька різних викладачів кафедри (з урахуванням категорії викладачів, наприклад, асистент не може читати лекції, а професор ніколи не буде проводити лабораторні роботи). Викладач може проводити заняття з однієї або декількох дисциплін для студентів як свого, так і інших факультетів. Відомості про проведені іспити і заліки збираються деканатом.

Після закінчення навчання студент виконує дипломну роботу,

керівником якої є викладач з кафедри, що відноситься до того ж факультету, де навчається студент, при цьому викладач може керувати декількома студентами.

Види запитів в інформаційній системі:

1. Отримати перелік і загальне число студентів визначених груп, або визначеного курсу (курсів) факультету повністю, за статевою ознакою, роком народження, віком, ознакою наявності дітей, за ознакою отримання і розміром стипендії.

2. Отримати список і загальне число викладачів визначених кафедр або визначеного факультету повністю, або визначених категорій (асистенти, доценти, професори і т.д.) за статевою ознакою, роком народження, віком, ознакою наявності та кількості дітей, розміром заробітної плати, аспірантів, які захистили кандидатські, кандидатів наук - докторські дисертації в визначений період.

3. Отримати перелік і загальне число тем кандидатських і докторських дисертацій, захищених співробітниками визначеної кафедри, або визначеного факультету.

4. Отримати перелік кафедр, які проводять заняття у визначеній групі або на визначеному курсі вказаного факультету в визначеному семестрі, або за вказаний період.

5. Отримати список і загальне число викладачів, які проводили (проводять) заняття з вказаної дисципліни у визначеній групі, або на визначеному курсі вказаного факультету.

6. Отримати перелік і загальне число викладачів, які проводили (проводять) лекційні, семінарські та інші види занять у визначеній групі, або на визначеному курсі вказаного факультету в визначеному семестрі, або за вказаний період.

7. Отримати список і загальне число студентів визначених груп, які здали залік, або іспит з визначеної дисципліни з вказаною оцінкою.

8. Отримати список і загальне число студентів визначених груп або

вказаного курсу визначеного факультету, які здали визначену сесію на відмінно, без трійок, без двійок.

9. Отримати перелік викладачів, які беруть (брали) участь у іспитах у визначених групах, із визначених дисциплін, у визначеному семестрі.

10. Отримати список студентів визначених груп, або яким заданий викладач поставив деяку оцінку за іспит з певних дисциплін, у визначених семестрах, за деякий період.

11. Отримати список студентів і тем дипломних робіт, що виконувались ними на визначеній кафедрі, або у визначеного викладача.

12. Отримати список керівників дипломних робіт з визначеної кафедри, або факультету повністю і окремо за деякими категоріями викладачів.

13. Отримати навантаження викладачів (назва дисципліни, кількість годин), його обсяг на окремі види занять і загальне навантаження у визначеному семестрі для конкретного викладача, або для викладачів визначеної кафедри.

5.3.2. Інформаційна система проектної організації

Опис предметної галузі. Проектна організація представлена наступними категоріями працівників: конструктори, інженери, техніки, лаборанти, інший обслуговуючий персонал. Кожна категорія може мати властиві тільки їй атрибути. Наприклад, конструктор характеризується числом авторських свідоцтв, техніки - обладнанням, яке вони можуть обслуговувати, інженер або конструктор можуть керувати договором або проектом і т.д. Співробітники розділені на відділи, що керуються начальником так, що кожен співробітник числиться тільки в одному відділі.

В рамках укладених проектною організацією договорів з замовниками виконуються різного роду проекти, причому за одним договором може виконуватися більше одного проекту, і один проект може виконуватися для кількох договорів. Сумарна вартість договору визначається вартістю всіх проектних робіт, які виконуються для цього договору. Кожен договір і

проект має керівника і групу співробітників, що виконують цей договір або проект, причому це можуть бути співробітники не тільки одного відділу. Проекти виконуються з використанням різного обладнання, частина якого приписана до окремих відділів, а частина є колективною власністю проектної організації, при цьому в процесі роботи обладнання може передаватися з відділу у відділ. Для виконання проекту обладнання надається групі, що працює над проектом, якщо це обладнання не використовується в іншому проекті.

Для виконання ряду проектів підрядна організація може залучати субпідрядні організації, передаючи їм обсяги робіт.

Ведеться облік кадрів, облік виконання договорів і проектів, вартісний облік усіх виконаних робіт.

Види запитів в інформаційній системі:

1. Отримати дані про склад визначеного відділу або всієї організації повністю, за визначеною категорією працівників, за віковим складом.

2. Отримати перелік керівників відділів.

3. Отримати перелік договорів або проектів, які виконуються в даний момент або в період визначеного інтервалу часу.

4. Отримати інформацію про те, які проекти виконуються (виконувалися) в рамках визначеного договору і які договори підтримуються визначеними проектами.

5. Отримати дані про вартість виконаних договорів (проектів) протягом визначеного періоду часу.

6. Отримати дані про розподіл обладнання на даний момент або на деяку визначену дату.

7. Отримати відомості про використання обладнання визначеними проектами (договорами).

8. Отримати відомості про участь вказаного співробітника або категорії співробітників у проектах (договорах) за певний період часу.

9. Отримати перелік і вартість робіт, виконаних субпідрядними

організаціями.

10. Отримати дані про чисельність і склад співробітників в цілому і за окремими категоріями, які беруть участь в визначеному проекті.

11. Отримати дані про ефективність використання обладнання (обсяги проектних робіт, виконаних з використанням того чи іншого обладнання).

12. Отримати відомості про ефективність договорів (вартість договорів співвіднесення з витраченим часом або вартість з урахуванням залучених людських ресурсів).

13. Отримати дані про чисельність і склад співробітників в цілому і за окремими категоріями, які беруть участь в проектах за вказаний період часу.

14. Отримати відомості про ефективність проектів (вартість договорів, співвіднесення з витраченим часом або вартість з урахуванням залучених людських ресурсів).

5.3.3. Інформаційна система бібліотечного фонду міста

Опис предметної галузі. Бібліотечний фонд міста складають бібліотеки, розташовані на території міста. Кожна бібліотека включає в себе абонементи і читальні зали. Користувачами бібліотек є різні категорії читачів: студенти, науковці, викладачі, школярі, робітники, пенсіонери та інші жителі міста. Кожна категорія читачів може володіти непересічними характеристиками-атрибутами: для студентів це назва навчального закладу, факультет, курс, номер групи, для науковця - назва організації, наукова тема і т.ін. Кожен читач, будучи зареєстрованим, в одній з бібліотек, має доступ до всього бібліотечного фонду міста.

Бібліотечний фонд (книги, журнали, газети, збірники статей, збірки віршів, дисертації, реферати, збірники доповідей та тез доповідей і т.ін.) Розміщений в залах-сховищах різних бібліотек на певних місцях зберігання (номер залу, стелажа, полиці) і ідентифікується номенклатурними номерами. При цьому існують різні правила щодо тих чи інших видань: якісь підлягають тільки читанню в читальних залах бібліотек, для тих, що

видаються, може бути встановлений різний термін видачі і т.д. З одного боку, бібліотечний фонд може поповнюватися, з іншого, - з плином часу відбувається його списання.

Твори авторів, що складають бібліотечний фонд, також можна розділити на різні категорії, які характеризуються власним набором атрибутів: підручники, повісті, романи, статті, вірші, дисертації, реферати, тези доповідей і т.д.

Співробітники бібліотеки, що працюють в різних залах різних бібліотек, ведуть облік читачів, а також облік розміщення та видачі літератури.

Види запитів в інформаційній системі:

1. Отримати список читачів із заданими характеристиками: студентів визначеного навчального закладу, факультету, науковців з певної тематики і т.ін.

2. Видати перелік читачів, на руках у яких перебуває визначений твір.

3. Отримати список читачів, на руках у яких знаходиться вказане видання (книга, журнал тощо).

4. Отримати перелік читачів, які протягом визначеного проміжку часу отримували видання з деяким твором і назвою цього видання.

5. Видати список видань, які протягом деякого часу отримував визначений читач з фонду бібліотеки, де він зареєстрований.

6. Отримати перелік видань, якими протягом деякого часу користувався вказаний читач з фонду бібліотеки, де він не зареєстрований.

7. Отримати список літератури, яка в даний момент видана з певною полиці деякої бібліотеки.

8. Видати список читачів, які протягом означеного періоду були обслужені визначеним бібліотекарем.

9. Отримати дані про роботу бібліотекарів (число обслужених читачів у визначений період часу).

10. Отримати список читачів з простроченим терміном літератури.

11. Отримати перелік визначеної літератури, яка надійшла (була списана) протягом деякого періоду.

12. Видати список бібліотекарів, які працюють у визначеному читальному залі деякої бібліотеки.

13. Отримати список читачів, які не відвідували бібліотеку протягом визначеного часу.

14. Отримати список інвентарних номерів і назв з бібліотечного фонду, в яких міститься вказаний твір.

15. Видати список інвентарних номерів і назв з бібліотечного фонду, в яких містяться твори визначеного автора.

16. Отримати список найпопулярніших творів.

5.3.4. Інформаційна система аптеки

Опис предметної галузі. Аптека продає медикаменти і виготовляє їх за рецептами. Ліки можуть бути різних типів: готові ліки (таблетки, мазі, настоянки) і виготовлені аптекою (мікстури, мазі, розчини, настойки, порошки).

Різниця у типах ліків відбивається у різному наборі атрибутів, що їх характеризують. Мікстури і порошки виготовляються тільки для внутрішнього застосування, розчини для зовнішнього, внутрішнього застосування і для змішування з іншими ліками та мазі тільки для зовнішнього застосування. Ліки різні також за способом приготування і за часом приготування. Порошок і мазі виготовляються змішуванням різних компонентів. При виготовленні розчинів і мікстур інградієнти не тільки змішують, але і відстоюють з подальшою фільтрацією ліків, що збільшує час виготовлення.

В аптеці існує довідник технологій приготування різних ліків. У ньому вказуються: ідентифікаційний номер технології, назву ліків і сам спосіб приготування. На складі на всі медикаменти встановлюється критична норма, тобто коли будь-які речовини на складі менше критичної норми, то

складаються заявки на дані речовини і їх в терміновому порядку привозять з оптових складів медикаментів.

Для виготовлення аптекою ліків, хворий повинен принести рецепт від лікаря. У рецепті повинно бути вказано: ПІБ, підпис і печатка лікаря, ПІБ, вік та діагноз пацієнта, також кількість ліків і спосіб застосування. Хворий віддає рецепт реєстратору, він приймає замовлення і дивиться, чи є компоненти, що замовляється ліки. Якщо не всі компоненти є в наявності, то робить заявки на оптові склади ліків і фіксує ПІБ, телефон та адресу необслушеного покупця, щоб повідомити йому, коли доставлять потрібні компоненти. Такий хворий поповнює довідник замовлень - це ті замовлення, які знаходяться в процесі приготування, з позначкою, що не всі компоненти є для замовлення. Якщо всі компоненти є, то вони резервуються для ліків хворого. Покупець сплачує ціну ліків, йому повертається рецепт з позначкою про час виготовлення. Хворий також поповнює довідник замовлень у виробництві. У призначений час хворий приходить і за тим самим рецептом отримує готові ліки. Такий хворий поповнює список виконаних замовлень.

Ведеться статистика за обсягами використовуваних медикаментів. Через певний проміжок часу проводиться інвентаризація складу. Це робиться для того, щоб визначити, чи є ліки з критичною нормою, або вийшов термін зберігання або є нестача ліків.

Види запитів в інформаційній системі:

1. Отримати відомості про покупців, які не прийшли забрати своє замовлення в призначений їм час і загальне їх число.
2. Отримати перелік і загальне число покупців, які чекають прибуття на склад потрібних їм медикаментів в цілому і за визначеною категорією медикаментів.
3. Отримати перелік десяти найбільш часто використовуваних медикаментів в цілому і визначеної категорії медикаментів зокрема.
4. Отримати який обсяг визначених речовин використаний за вказаний період.

5. Отримати перелік і загальне число покупців, які замовляли певні ліки або певні типи ліків за даний період.

6. Отримати перелік і типи ліків, які досягли своєї критичної норми або закінчилися.

7. Отримати перелік ліків з мінімальним запасом на складі в цілому і за визначеною категорією медикаментів.

8. Отримати повний перелік і загальне число замовлень, що знаходяться у впровадженні.

9. Отримати повний перелік і загальне число препаратів потрібних для замовлень, які перебувають у впровадженні.

10. Отримати всі технології приготування ліків визначених типів, конкретних ліків, ліків, що знаходяться у довіднику замовлень у виробництві.

11. Отримати відомості про ціни на вказані ліки в готовому вигляді, про обсяг і ціни на всі компоненти, потрібні для цих ліків.

12. Отримати відомості про замовлення клієнтів на медикаменти певного типу, що найбільш часто роблять пацієнти, на конкретні медикаменти.

13. Отримати відомості про конкретні ліки (їх тип, спосіб приготування, назви всіх компонентів, ціни, їх кількість на складі).

5.3.5. Інформаційна система міської телефонної мережі

Опис предметної галузі. МТС є розгалуженою мережею локальних АТС. АТС підрозділяються на міські, відомчі та установчі та, можливо, мають характерний тільки для цієї групи набір атрибутів. У кожній АТС є свої абоненти. У абонента може стояти телефон одного з трьох типів: основний, паралельний або спарений. За кожним абонентом (у нього є прізвище, ім'я, по батькові, стать, вік і т.д.) закріплений свій номер телефону, причому у декількох абонентів може бути один і той же номер (при паралельному або спареному телефоні). Кожному номеру телефону відповідає адреса (індекс, район, вулиця, будинок, квартира), причому

паралельні або спарені телефони обов'язково повинні перебувати в одному будинку.

Всі телефони міської АТС мають вихід на міжмісто, але для конкретного абонента він може бути або відкритий, або закритий з якої-небудь причини (відключений за бажанням абонента, за несплату і т.п.). Відомчі та установчі АТС мають свою внутрішню мережу телефонів. Відомості про міжміські переговори збираються і аналізуються на МТС.

Абоненти зобов'язані платити абонентську плату. Плата повинна вноситися щомісяця до 20-го числа. При несплаті після письмового повідомлення протягом двох діб відключається абонент. При заборгованості за міжміські розмови і несплаті після письмового повідомлення проводиться відключення тільки можливості виходу на міжмісто. Включення того і (або) іншого проводиться при оплаті вартості включення, абонентської плати та пені.

Абонентів АТС можна поділити на простих і пільгових. До категорії пільговиків належать пенсіонери, інваліди тощо. Пільговики платять тільки 50% абонентської плати. У відповідності з усім цим (тип телефону, пільговик чи ні, чи є вихід на міжмісто) розраховується розмір абонентської плати.

На установку телефону існують черги: пільгова і звичайна. При підході черговості розглядається технічна можливість установки (наявність кабелю і вільного каналу, наявність вільних телефонних номерів).

У місті також існують громадські телефони і таксофони, розташовані за певними адресами.

Види запитів в інформаційній системі:

1. Отримати перелік і загальне число абонентів визначеної АТС повністю, тільки пільговиків, за віковою ознакою, за групою прізвищ.
2. Отримати перелік і загальне число вільних телефонних номерів на визначеній АТС, по всій МТС, за ознакою можливості установки телефону в даному районі.
3. Отримати перелік і загальне число боржників на визначеній АТС, по

всій МТС, по даному району, абонентів, які мають заборгованість вже більше тижня (місяця), за ознакою заборгованості (за міжміські, та (або) за абонентську плату, за розміром боргу).

4. Визначити АТС (будь-якого або конкретного типу), на якій найбільше (найменше) число боржників, найбільша сума заборгованості.

5. Отримати перелік і загальне число громадських телефонів і таксофонів у всьому місті, що належать визначеній АТС, за ознакою знаходження в даному районі.

6. Знайти відсоткове співвідношення звичайних і пільгових абонентів на визначеній АТС, по всій МТС, по даному району, за типами АТС.

7. Отримати перелік і загальне число абонентів визначеної АТС, по всій МТС, по даному району, за типами АТС, що мають паралельні телефони, тільки пільговиків, які мають паралельні телефони.

8. Визначити, чи є за даною адресою телефон, загальна кількість телефонів і (або) кількість телефонів з виходом на міжмісто, з відкритим виходом на міжмісто в даному будинку, на конкретній вулиці.

9. Визначити місто, з яким відбувається найбільша кількість міжміських переговорів.

10. Отримати повну інформацію про абонентів з заданим телефонним номером.

11. Отримати перелік спарених телефонів, для яких є технічна можливість замінити їх на звичайні (виділити додатковий номер).

12. Отримати перелік і загальне число внутрішніх телефонів на певній відомчій станції або АТС, з яких за певний період часу було здійснено менше певне числа зовнішніх дзвінків.

13. Отримати перелік і загальне число боржників на визначеній АТС, по всій МТС, по даному району, яким слід надіслати письмове повідомлення.

5.3.6. Інформаційна система фотоцентра

Опис предметної галузі. Фотоцентр має головний офіс і мережу філій і

кіосків прийому замовлень, розташованих за певними адресами. Філії та кіоски розрізняються за кількістю робочих місць. У кіосках здійснюється тільки прийом замовлень, тому кожен кіоск прикріплений до певної філії, в якій ці замовлення виконуються. У філіях є необхідне обладнання для друку фотографій. Філії та кіоски приймають замовлення на друк фотографій. У замовленні на друк вказується кількість фотографій з кожного кадру, загальна кількість фотографій, формат, тип паперу і терміновість виконання замовлення. При замовленні великої кількості фотографій надаються знижки. Термінові замовлення приймаються тільки в філіях, і вони мають ціну в два рази більшу, ніж звичайне замовлення. При придбанні дисконтної картки клієнт отримує значні знижки на друк фотографій.

Клієнтів можна розділити на професіоналів і любителів. Професіоналам, що приносить замовлення в одну і ту ж філію, можуть бути запропоновані персональні знижки. Фотомагазини і кіоски пропонують до продажу різні фототовари: фотоаксесуари, фотоапарати, альбоми та інші фототовари. Фотомагазини також пропонують додаткові види послуг: фотографії на документи, реставрацію фотографій, прокат фотоапаратів, художнє фото, надання послуг професійного фотографа.

Відомості про виконані замовлення і продаж різних фототоварів збираються і обробляються, і на основі цієї інформації робиться загальне замовлення на поставку витратних матеріалів (фотопапір, картриджі для друку і т. ін.), фототоварів і обладнання. Отримані товари і матеріали розподіляються відповідно до запитів по кіосках і магазинах. У фотоцентра може бути кілька постачальників, які спеціалізуються на різних поставках, або на постачанні фототоварів різних фірм.

Види запитів в інформаційній системі:

1. Отримати перелік і загальне число пунктів прийому замовлень на фотороботи по філіям, по кіосках прийому замовлень, в цілому по фотоцентру.

2. Отримати перелік і загальне число замовлень на фотороботи по

філіям, кіоскам прийому замовлень, в цілому по фотоцентру, що надійшли протягом деякого періоду часу.

3. Отримати перелік і загальне число замовлень (окремо простих і термінових) на окремі види фоторобіт за вказаною філією, кіоском прийому замовлень, що надійшли протягом деякого періоду часу.

4. Отримати суму виручки з замовлень (окремо простих і термінових) на окремі види фоторобіт за вказаною філією, кіоском прийому замовлень, що надійшли протягом деякого періоду часу.

5. Отримати кількість віддрукованих фотографій в рамках простих і термінових замовлень за вказаною філією, кіоском прийому замовлень, фотоцентром в цілому за певний період часу.

6. Отримати кількість віддрукованих фотографій в межах простих і термінових замовлень за вказаною філією, кіоском прийому замовлень, фотоцентром в цілому за певний період часу.

7. Отримати перелік постачальників в цілому по фотоцентру, постачальників окремих видів фототоварів, які зробили поставки в певний період, поставки певного обсягу.

8. Отримати список клієнтів в цілому по фотоцентру, клієнтів визначеної філії, що мають знижки, які зробили замовлення певного обсягу.

9. Отримати суму виручки від реалізації фототоварів в цілому по фотоцентру, за вказаною філією, проданих протягом деякого періоду часу.

10. Отримати перелік фототоварів і фірм, що їх виробляють, які користуються найбільшим попитом в цілому по фотоцентру, в визначеній філії.

11. Отримати перелік реалізованих фототоварів і обсяги їх реалізації в цілому по фотоцентру, за вказаною філією, проданих протягом деякого періоду часу.

12. Отримати перелік робочих місць фотоцентра в цілому і визначеного профілю - окремо.

5.3.7. Інформаційна система залізничної пасажирської станції

Опис предметної галузі. Працівників залізничної станції можна поділити на водіїв рухомого складу, диспетчерів, ремонтників рухомого складу, шляхів, касирів, працівників служби підготовки складів, довідкової служби та інших, які адміністративно відносяться кожен до свого відділу. Кожна з перерахованих категорій працівників має атрибути-характеристики, які визначаються професійною спрямованістю. У відділах існує розбиття працівників на бригади. Відділи очолюються начальниками, які являють собою адміністрацію залізничної станції. У функції адміністрації входить планування маршрутів, складання розкладів, формування кадрового складу залізничної станції. За кожним локомотивом закріплюється бригада. За локомотивами закріплюється бригада техніків-ремонтників, що виконує за графіком рейсовий і плановий техогляд, ремонт, технічне обслуговування. Водії локомотивів зобов'язані щороку проходити медогляд. Водії, що не пройшли медогляд переводяться на іншу роботу. Локомотив повинен своєчасно оглядатися ремонтниками і при необхідності ремонтуватися. Підготовка до рейсу включає в себе технічну частину (техогляд, дрібний ремонт) і обслуговуючу частину (прибирання вагонів, запас продуктів і т.п.).

У розкладі вказується тип поїзда (швидкий, пасажирський...), Номер поїзда, дні і час відправлення і прибуття, маршрут (початковий і кінцевий пункти призначення, основні вузлові станції), вартість квитка. Квитки на поїзд можна придбати заздалегідь або забронювати в залізничних касах. До відправлення поїзда, якщо є необхідність, квиток можна повернути. Відправлення поїздів може бути затримано через запізнення поїздів, погодних умов, технічних неполадок.

Залізничні маршрути можна розділити на наступні категорії: внутрішні, міжнародні, туристичні, спеціальні маршрути. Пасажири можуть здавати свої речі в багажне відділення.

Види запитів в інформаційній системі:

1. Отримати перелік і загальне число всіх працівників залізничної

станції за стажем роботи на станції, статевою ознакою, віком, ознакою наявності та кількості дітей, розміром заробітної плати.

2. Отримати загальне число працівників по всіх відділах, які обслуговують певний локомотив, за віком, зарплатою в бригаді.

3. Отримати перелік і загальне число водіїв локомотивів, що пройшли медогляд або не пройшли медогляд у визначений рік, за статевою ознакою, віком, розміром заробітної плати.

4. Отримати перелік і загальне число локомотивів, приписаних до залізничної станції, що знаходяться на ній в визначений час, за часом прибуття на станцію, за кількістю маршрутів обслуговування.

5. Отримати загальне число локомотивів, що пройшли плановий техогляд за певний період часу, відправлених у ремонт в означений час, за кількістю виконаних рейсів до ремонту, за віком локомотива.

6. Отримати перелік і загальне число поїздів на визначеному маршруті, за тривалістю маршруту, за ціною квитка.

7. Отримати перелік і загальне число скасованих рейсів повністю, у визначеному напрямку, за вказаним маршрутом.

8. Отримати перелік затриманих рейсів повністю, із визначеної причини, за вказаним маршрутом і кількість зданих квитків за час затримки.

9. Отримати перелік і середню кількість проданих квитків за вказаний інтервал часу на певні маршрути, за тривалістю маршруту, за ціною квитка.

10. Отримати перелік і загальне число маршрутів визначеної категорії, які прямують у певному напрямку.

11. Отримати перелік і загальне число пасажирів на визначеному рейсі, які виїхали у визначений день, які виїхали за кордон у визначений день, за ознакою здачі речей в багажне відділення, за статевою ознакою, за віком.

12. Отримати перелік і загальне число невикуплених квитків на визначеному рейсі, за день, за деяким маршрутом.

13. Отримати загальне число зданих квитків на вказаний рейс, день, маршрут.

ЛІТЕРАТУРА

1. Гармаш А.Н. Математические методы в управлении: учеб. пособие для студентов вузов / А.Н. Гармаш, И.В. Орлова. – М.: Вузовский учебник: ИНФРА-М, 2012. – 272 с.
2. Дудник І.М. Вступ до загальної теорії систем: Навчальний посібник/ І. М. Дудник. - К.: Кондор, 2009. - 205 с.
3. Электронный портал <http://simulation.in.ua>
4. Емельянов А.А. Компьютерная имитация экономических процессов / А.А. Емельянов, Е.А. Власова, Р.В. Дума и др.; под ред. А.А. Емельянова. – М.: Маркет ДС, 2010. – 464 с.
5. Качала В.В. Основы теории систем и системного анализа / В.В. Качала. – М.: Горячая линия-Телеком, 2007. – 216 с.
6. Кийкова Е.В. Имитационное моделирование: практикум для студ. вузов / Е.В. Кийкова, Е.Г. Лаврушина. – Владивосток: Изд-во ВГУЭС, 2005. – 100 с.
7. Колесов Ю.Б. Моделирование систем: практикум по компьютерному моделированию / Ю.Б. Колесов, Ю.Б. Сениченков. – СПб.: БХВ-Петербург, 2007. – 352 с.
8. Методические рекомендации по выполнению практических работ по курсу "Компьютерные методы проектирования" [Электронный ресурс]/ - Электрон. текстовые данные. - Саратов: Вузовское образование, 2013.- 186 с - <http://www.iprbookshop.ru/12807>.
9. Синенко С.А. Компьютерные методы проектирования [Электронный ресурс]: учебно-практическое пособие/ Синенко С.А., Славин А.М., Жадановский Б.В. - Электрон. текстовые данные. - М.: МГСУ, Ай Пи Эр Медиа, ЭБС АСВ, 2015. - 138 с. - <http://www.iprbookshop.ru/40571>.
10. Ситник В.Ф., Орленко Н.С. Імітаційне моделювання: Навч.-метод. посібник для самост. вивч. дисц. - К.: КНЕУ, 1999. – 208 с [Электронный ресурс] - режим доступа: <http://studentam.kiev.ua/content/view/701/94/>.
11. Старіш О.Г. Системологія: Підручник / О.Г. Старіш. - К.: Центр навчальної літератури, 2005. – 232 с.
12. Томашевський В.М. Моделювання систем: Підручник / В. М. Томашевський : За заг. ред. академіка НАН України М.З. Згуровського - К.: Видавнича група ВНУ, 2005. - 352 с.

ЗМІСТ

1. МЕТОДОЛОГІЇ СТРУКТУРНОГО МОДЕЛЮВАННЯ.....	3
1.1. Діаграми потоків даних. Нотація Йордана-де Марко. Нотація Гейне-Сарсона.....	4
1.2. Методологія IDEF	9
2. МЕТОДОЛОГІЇ ІНФОРМАЦІЙНОГО МОДЕЛЮВАННЯ.....	15
2.1. Нотація Чена	16
2.2. Нотація Мартіна	17
2.3. Нотація IDEFIX	18
2.4. Нотація Баркера.....	20
2.5. Мова інфологічного моделювання	21
3. РОЗРОБКА ФУНКЦІОНАЛЬНОЇ МОДЕЛІ ПРЕДМЕТНОЇ ГАЛУЗІ В DESIGN/IDEF	22
3.1. Запуск IDEF0 моделювання	22
3.2. Створення нової IDEF0-моделі.....	22
3.3. Налаштування середовища проектування	24
3.4. Створення та редагування функціональних блоків.....	25
3.5. Додавання тексту до функціонального блоку	25
3.6. Створення та редагування ярликів	26
3.7. Створення дуг	26
3.8. Створення підписів дуг, що з'єднують блоки	27
3.9. Декомпозиція блоків.....	28
3.10. Переміщення між сторінками IDEF0-моделі.....	28
3.11. Заповнення глосарію	28
3.12. Перевірка синтаксису IDEF0-моделі	31
3.13. Збереження IDEF0-моделі	32
3.14. Приклад IDEF0-моделі «Виконати розрахунково-графічну роботу (РГР).....	32
4. РОЗРОБКА ІНФОРМАЦІЙНОЇ МОДЕЛІ ПРЕДМЕТНОЇ ГАЛУЗІ В DESIGN/IDEF.....	35
4.1. Основні компоненти IDEF1X- моделі.....	35

4.1.1.	Сутність.....	35
4.1.2.	Атрибути.....	36
4.1.3.	Відношення (зв'язок).....	40
4.2.	Розробка концептуальної схеми бази даних в Design/IDEF.....	44
4.2.1.	Запуск IDEF1X моделювання.....	44
4.2.2.	Створення нової IDEF1X-моделі.....	44
4.2.3.	Створення об'єктів IDEF1X класу.....	45
4.3.	Збереження IDEF1X-моделі.....	49
4.4.	Генерація sql-скрипта для створення бази даних на IDEF1X-моделі.....	63
4.5.	Приклад IDEF1X-моделі.....	49
5.	ЛАБОРАТОРНИЙ ПРАКТИКУМ.....	51
5.1.	Лабораторна робота №1: Методика побудови функціональної моделі предметної галузі для проектування автоматизованої системи управління.....	51
5.1.1.	Послідовність виконання лабораторної роботи.....	51
5.1.2.	Вимоги до звіту.....	52
5.1.3.	Контрольні питання.....	52
5.2.	Лабораторна робота №2: Методика побудови інформаційної моделі предметної галузі для проектування автоматизованої інформаційної системи.....	53
5.2.1.	Послідовність виконання лабораторної роботи.....	53
5.2.2.	Вимоги до звіту.....	54
5.2.3.	Контрольні питання.....	55
5.3.	Варіанти завдань до лабораторного практикуму.....	55
5.3.1.	Інформаційна система Вузу.....	56
5.3.2.	Інформаційна система проектної організації.....	58
5.3.3.	Інформаційна система бібліотечного фонду міста.....	60
5.3.4.	Інформаційна система аптеки.....	62
5.3.5.	Інформаційна система міської телефонної мережі.....	64
5.3.6.	Інформаційна система фотоцентра.....	66
5.3.7.	Інформаційна система залізничної пасажирської станції.....	69
	ЛІТЕРАТУРА.....	71