

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»  
факультет програмної інженерії та бізнесу  
кафедра інженерії програмного забезпечення

## Лабораторна робота № 7

з дисципліни «Основи програмної інженерії»

*назва дисципліни*

на тему: АНАЛІЗ БІЗНЕС-ПРОЦЕСІВ З ВИКОРИСТАННЯМ IDEF0 з використанням  
онлайн сервісу Draw.io

Виконав: студент 1 курсу групи № 612п  
освітньої програми 121 інженерія  
програмного забезпечення

(шифр і назва ОП)

Ващенко А.А.

(прізвище й ініціали студента)

Прийняв: к.е.н., доцент каф. 603

Дем'яненко О.С

(посада, науковий ступінь, прізвище й ініціали)

Кількість балів:

Харків 2023

**Мета роботи:** ознайомитися з функціональною методикою проектування IDEF0 на прикладі опису процесу в середовищі DrawIO. Викласти роботу для зберігання в систему контролю версій Github.

## Завдання

1. Створити новий проект в середовищі DrawIO. , (Також рекомендується встановити альбомну орієнтацію сторінки).
2. Відповідно до варіанта завдання табл. 2 розробити IDEF0 діаграму з рівнем декомпозиції не менше трьох (див. приклад).
3. Розмістити та зберігти роботу на Github. Зробити скрін шоти та вставити їх в звіт.
4. Скласти звіт.

## Зміст звіту

1. Постановка задачі.
2. Словесний опис бізнес-процесу відповідно до варіанта.
3. IDEF0 діаграми.
4. Викладення скрин шотів .
5. Короткий опис кожної з діаграм.
6. Висновки.

## Теоретичні відомості

Для моделювання складних систем існують добре обкатані методології та стандарти. До них відносяться, зокрема, методології сімейства IDEF, за допомогою яких можна ефективно представляти і аналізувати моделі діяльності широкого спектру складних систем в різних розрізах. При цьому глибина дослідження процесів у системі визначається самим розробником, що дозволяє не перевантажувати створювану модель зайвими даними.

Методологію IDEF0 можна вважати кінцевим етапом розвитку добре відомої графічної мови опису функціональних систем SADT (Structured Analysis and Design Technique).

Стандарт IDEF0 був розроблений в 1981 році в рамках великої програми автоматизації промислових підприємств ICAM (Integrated Computer Aided Manufacturing), запропонованої департаментом Військово-Повітряних Сил

США. Сімейство стандартів IDEF успадкувало своє позначення від назви цієї програми (IDEF ICAM DEFinition).

Під час реалізації програми виникла необхідність розробити нові методи аналізу процесів взаємодії в промислових системах. Крім вдосконаленого набору функцій для опису бізнес-процесів, однією з вимог стало наявність ефективної методології взаємодії в рамках "аналітик-фахівець". Новий метод повинен був забезпечити групову роботу над створенням моделі, з безпосередньою участю всіх аналітиків і фахівців, зайнятих в рамках проекту. Так і виникла методологія функціонального моделювання IDEF0. З 1981 року стандарт IDEF0 зазнав кілька незначних змін, в основному обмеження характеру. Остання його редакція була випущена в грудні 1993 року Національним Інститутом Стандартів і Технологій США (NIST).

### Функціональний блок

Графічна мова IDEF0 дивно проста і гармонійна. В основі методології лежать чотири основних поняття, перше з яких - поняття функціонального блока (Activity Box). Функціональний блок графічно зображується у вигляді прямокутника (рис. 4), і, за вимогами стандарту, його назва повинна містити дієслівну форму ( "виробляти послуги", а не "виробництво послуг").

Кожна з чотирьох сторін функціонального блока має своє значення: верхня сторона - "Керування" (Control), ліва сторона - "Вхід" (Input), права - "Вихід" (Output), нижня сторона - "Механізм" (Mechanism). Кожному функціональному блоку в рамках системи привласнюється унікальний ідентифікаційний номер.

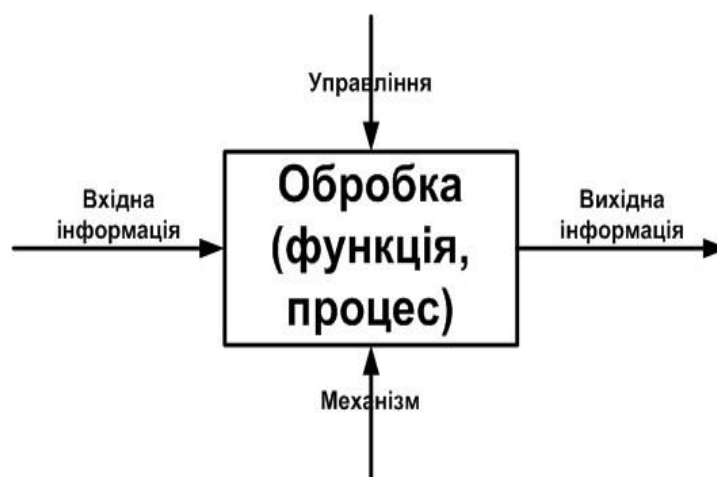


Рис. 4. Функціональний блок

За допомогою інтерфейсних дуг відображаються різні об'єкти, в тій чи іншій мірі визначають процеси, що відбуваються в системі. Такими об'єктами

можуть бути елементи реального світу (деталі, вагони, співробітники і т. д.), або потоки даних й інформації (документи, дані, інструкції і т. д.). Залежно від того, куди підходить інтерфейсна дуга, її називають "вхідною", "вихідною" або "керуючою". "Джерелом" (початком) кожної дуги може бути тільки вихідна сторона функціонального блока, а "приймачем" (кінцем) – будь-яка з трьох інших сторін.

На рис. 5 зображений функціональний блок "написання дипломної роботи". У реальному процесі є методичні рекомендації для написання дипломної роботи. Може здатися, що і диплом, і документ з рекомендаціями є вхідними об'єктами, але це не так. Насправді, диплом обробляється за правилами, які відображені в методичних рекомендаціях, які зображуються керуючою інтерфейсною дугою.



Рис.5. Приклад опису процесу написання диплому.

Однією з найбільш важливих особливостей методології IDEF-SADT є поступове введення все більш детальних уявлень моделі системи у міру розробки окремих діаграм. Побудова моделі IDEF-SADT починається з отримання всієї системи у вигляді найпростішої діаграми, що складається з одного блоку процесу і стрілок ICOM, службовців для зображення основних видів взаємодії з об'єктами поза системою. Оскільки вихідний процес представляє всю систему як єдине ціле, дане подання є найбільш загальним і підлягає подальшій декомпозиції.

Декомпозиція є умовним прийомом, що дозволяє уявити систему у вигляді, зручному для сприйняття, і оцінити її складність. В результаті декомпозиції підсистеми за певними ознаками виділяються окремі структурні елементи та зв'язки між ними. Декомпозиція служить засобом, що дозволяє уникнути

труднощів у розумінні системи. Глибина декомпозиції визначається складністю і розмірністю системи, а також цілями моделювання.

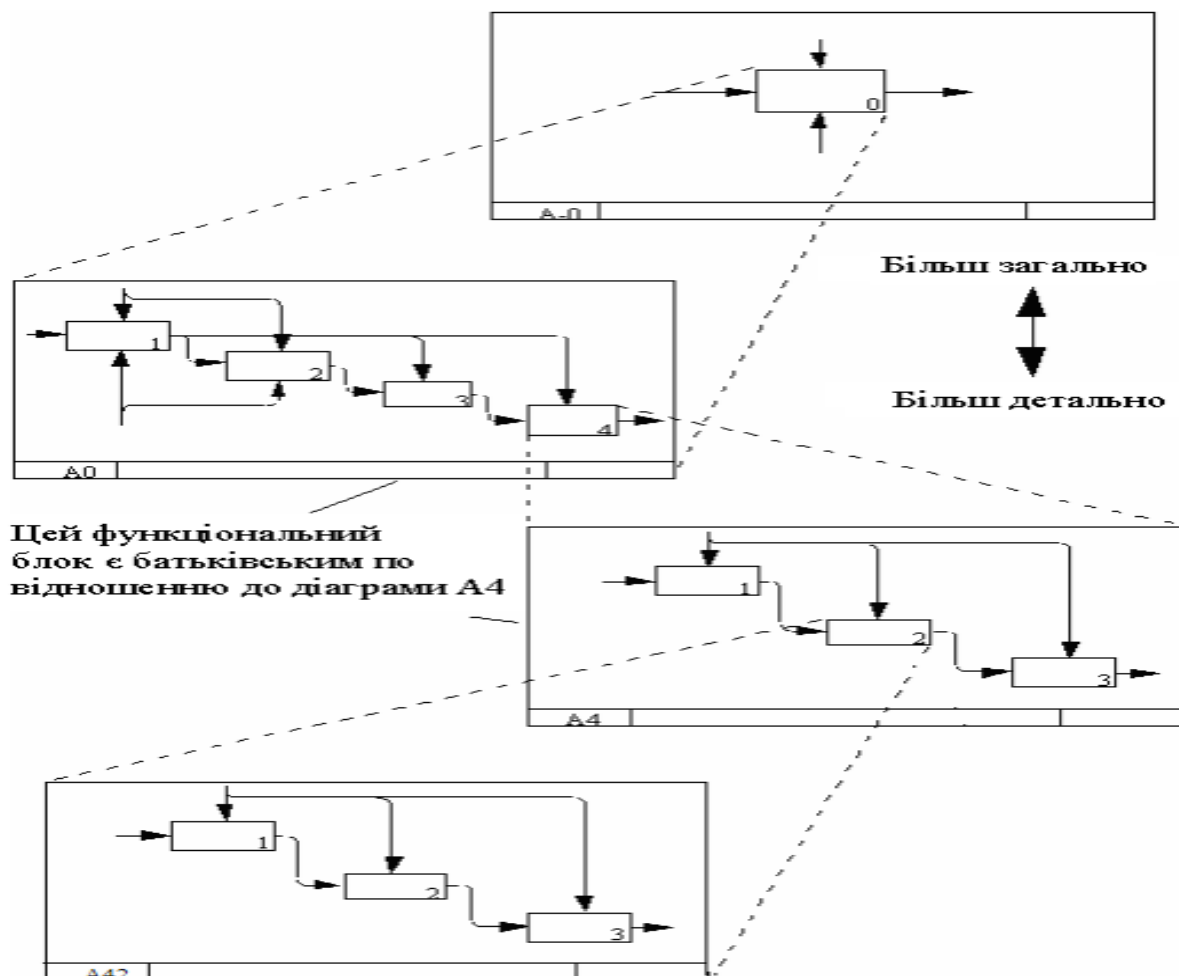


Рис. 6. Декомпозиція функціональних блоків

Слід звернути увагу на взаємозв'язок нумерації функціональних блоків і діаграм: кожен блок має свій унікальний порядковий номер на діаграмі (цифра в правому нижньому кутку прямокутника), а позначення під правим кутом вказує на номер дочірньої для цього блока діаграми. Відсутність такого позначення свідчить про те, що декомпозиції для цього блока не існує.

Часто окремі інтерфейсні дуги не варто розглядати в дочірніх діаграмах нижче або вище певного рівня. Наприклад, інтерфейсну дугу, яка зображує "деталь" на вході у функціональний блок "Обробити на токарному верстаті", немає сенсу відображати на діаграмах більш високих рівнів - це буде тільки перевантажувати їх і робити складними для сприйняття. Також буває необхідно позбутися від окремих "концептуальних" інтерфейсних дуг і не деталізувати їх глибше деякого рівня.

Для вирішення подібних завдань в стандарті IDEF0 передбачено поняття тунелювання. Символ "тунелю" (Arrow Tunnel) – дві круглі дужки навколо початку інтерфейсної дуги – позначає, що дуга була успадкована від функціонального батьківського блока і з'явилася тільки на цій діаграмі. "Тунель" навколо кінця (стрілки) інтерфейсної дуги в безпосередній близькості від блока-приймача означає, що в дочірньої по відношенню до цього блока діаграмі ця дуга відображатися та розглядатися не буде. Як правило, окремі об'єкти і відповідні їм інтерфейсні дуги "прибираються" на проміжних рівнях ієрархії. У цьому випадку вони спочатку "занурюються в тунель", а потім "повертаються з тунелю".

### Обмеження складності IDEF0-діаграм

Як правило, IDEF0-моделі містять складну і концентровану інформацію. Щоб зменшити їх перевантаженість і зробити легким для читання, відповідний стандарт рекомендує розміщувати:

- від трьох до шести функціональних блоків на діаграмі. Верхня межа (шість) змушує розробника використовувати ієрархії при описі складних предметів, а нижня межа (три) гарантує, що на відповідній діаграмі досить деталей, щоб виправдати її створення;
- чотири інтерфейсні дуги, відповідні до одного функціонального блока (або виходять з нього).

Суворо дотримуватися цих обмежень необов'язково, але, як показує досвід, вони дуже корисні в реальній роботі.

Зазвичай IDEF0-моделі несуть в собі складну і концентровану інформацію, і для того, щоб обмежити їх перевантаженість і зробити легким для читання, у відповідному стандарті прийняті відповідні обмеження складності:

- Обмеження кількості функціональних блоків на діаграмі трьома-шістьма. Верхня межа (шість) змушує розробника використовувати ієрархії при описі складних предметів, а нижня межа (три) гарантує, що на відповідній діаграмі досить деталей, щоб виправдати її створення;
- Обмеження кількості відповідних до одного функціонального блоку (що виходять з одного функціонального блоку) інтерфейсних дуг чотирма.

Приклад IDEF0 діаграм для процесу «Виконання лабораторної роботи» відображено на рис. 7– 10.

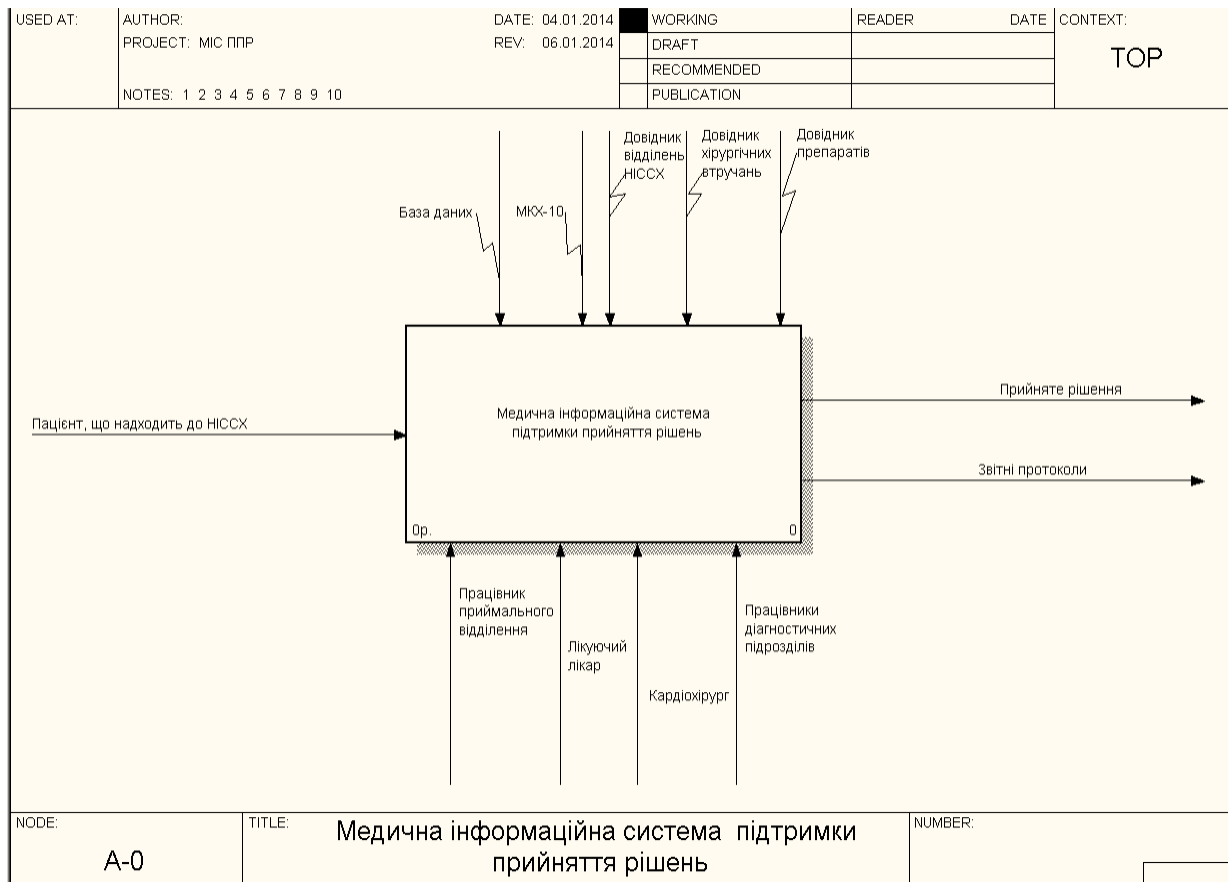


Рис.7. Модель бізнес-процесів медичної інформаційної системи підтримки прийняття рішень кардіологічного відділення (IDEF0)

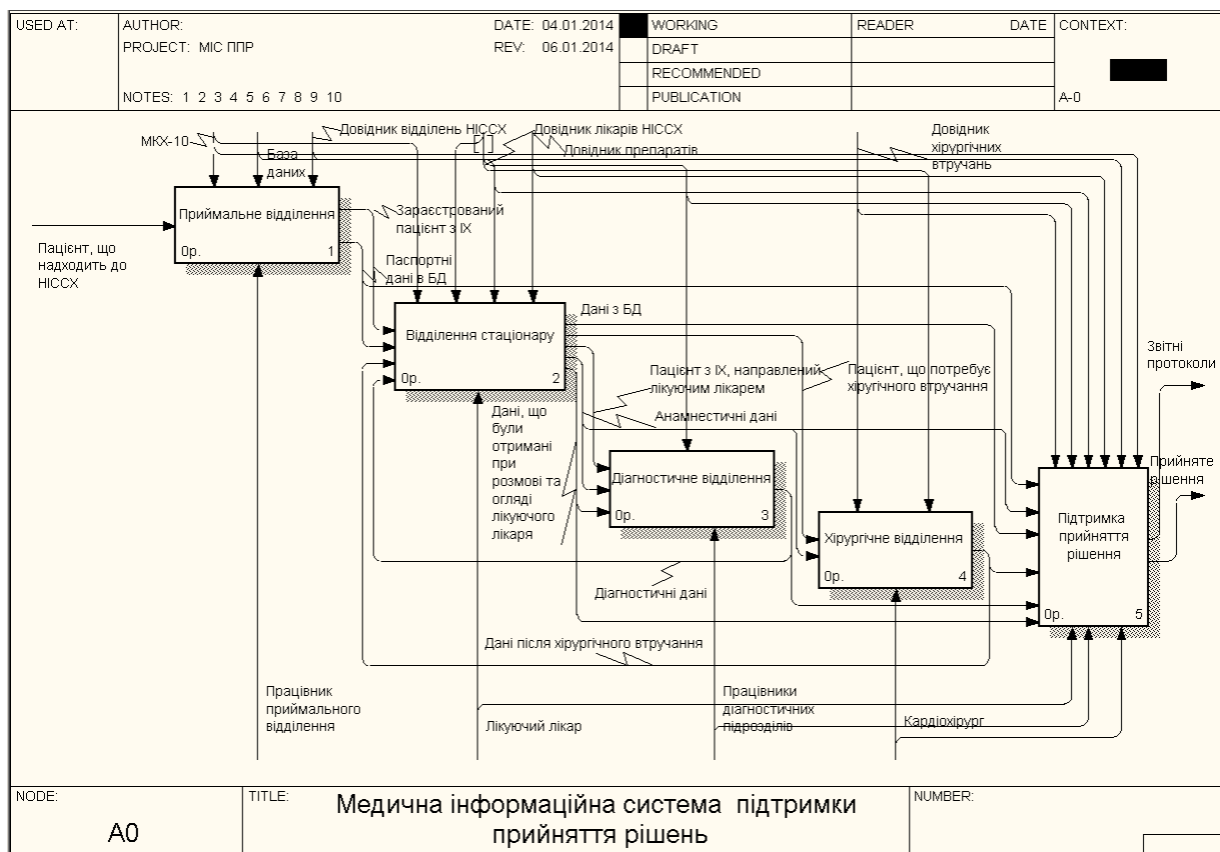


Рис.8. Перший рівень декомпозиції (IDEF0)



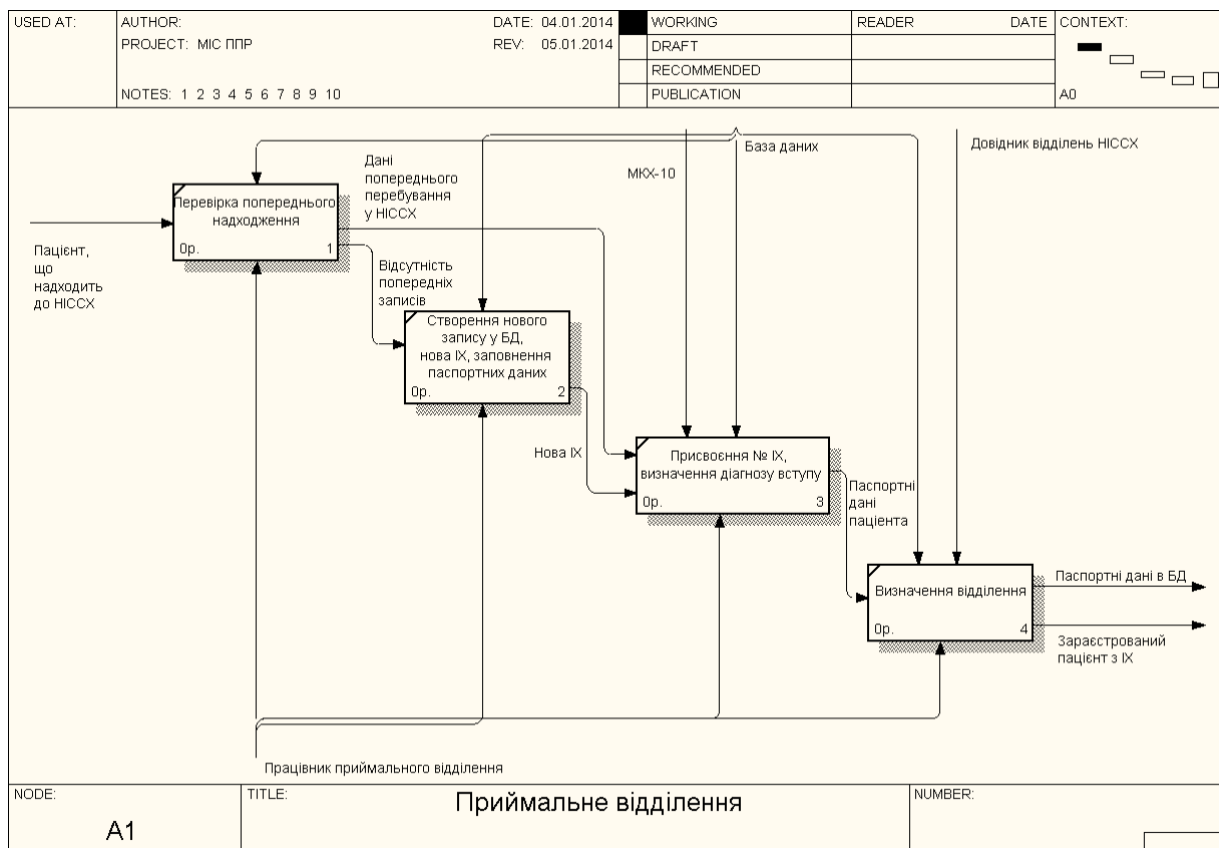


Рис.9. Діаграма декомпозиції IDEF0 (аналіз бізнес-процесів при впровадженні МІСППР)

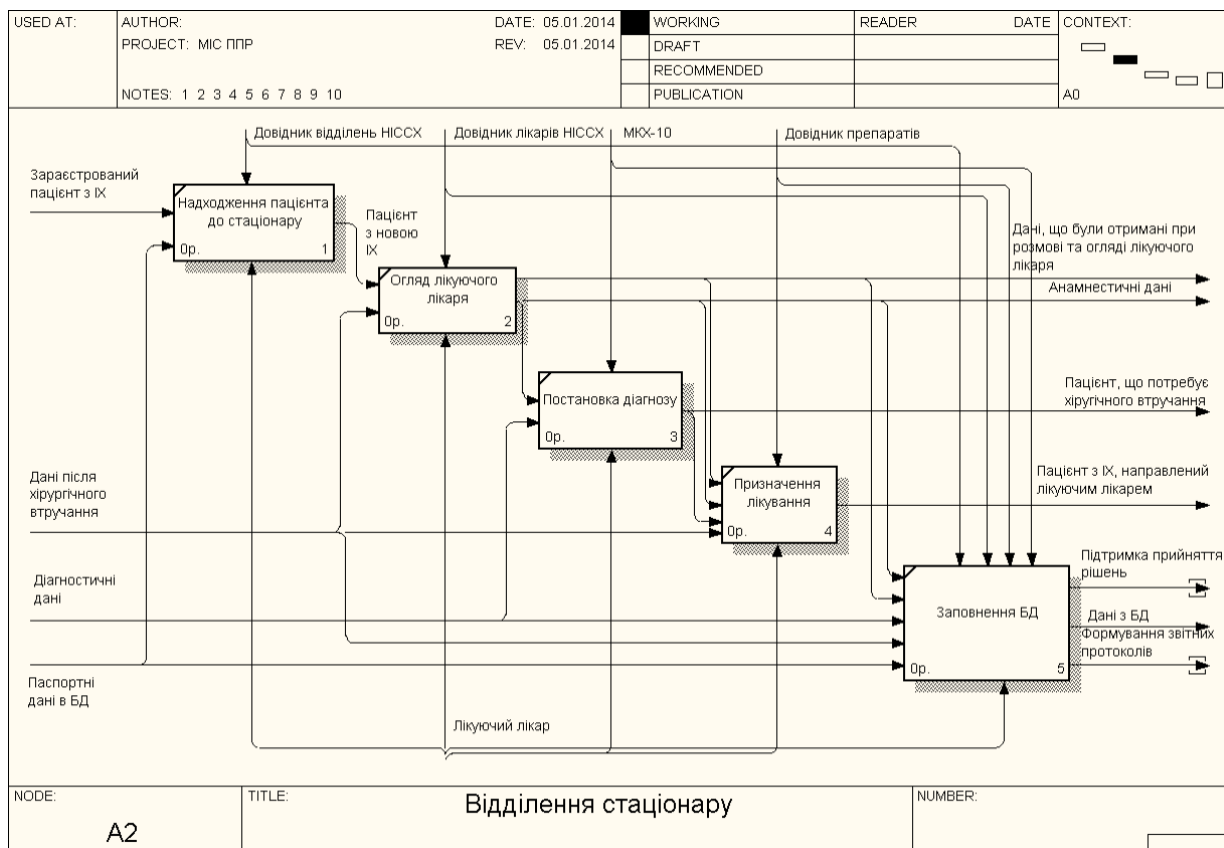


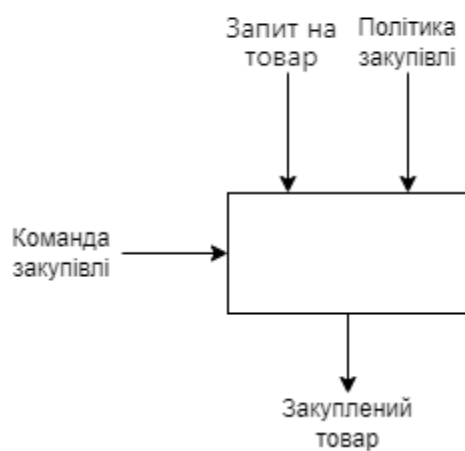
Рис.10. Діаграма декомпозиції IDEF0 (аналіз бізнес-процесів при впровадженні МІСППР)

## Варіант 2

### “Закупівля товару”

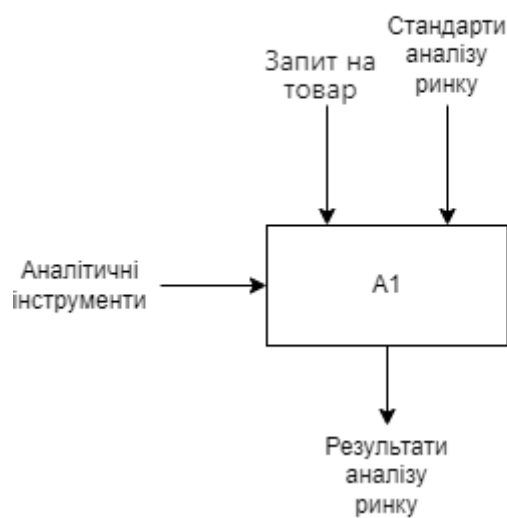
#### Виконання роботи

#### “Закупівля товару”



Мал. 11. головний блок

### “Аналіз ринку”



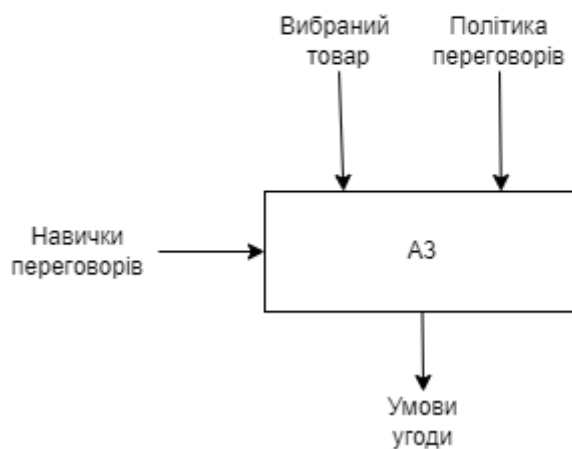
Мал. 12. А1: Аналіз ринку

### “Вибір товару”



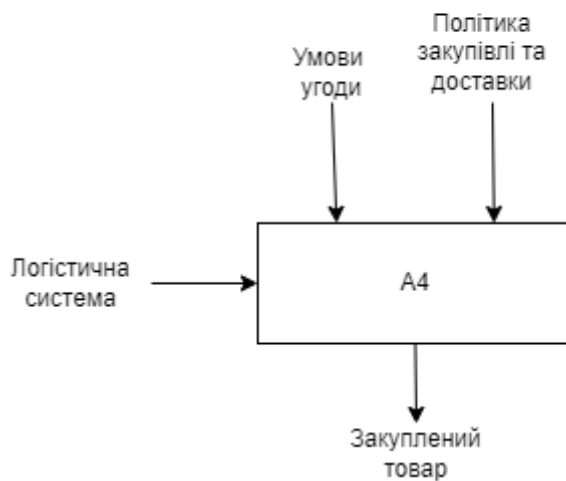
Мал. 13. А2: “Вибір товару”

### “Переговори з постачальником”



Мал. 14. A3: “Переговори з постачальником”

### “Закупівля та доставка товару”



Мал. 15. A4: “Закупівля та доставка товару”

## Висновок

В ході виконання даної лабораторної роботи я зміг навчитися робити IDEF0-діаграми та вдосконалив свої навички з роботи на сайті DrawIo

