

/Вісник ДУ "Львівська політехніка". 2000. 3. Литвак Б.Г. Экспертная информация: Методы получения и анализа. -М., 1982. 4. Э.А. Трахтенгерц. Компьютерная поддержка принятия решений: научн.практ. издание. Сер. "Информатизация России на пороге XXI века." М., 1998.

УДК 681.518:681.327.8

Є.В.Буров

НУ "Львівська політехніка", кафедра інформаційних систем та мереж

АВТОМАТИЗАЦІЯ АНАЛІЗУ ФУНКЦІОНАЛЬНОЇ СТРУКТУРИ РОЗПОДІЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

© Є.В.Буров, 2000

The functional structure of distributed information system remains an important media for both system designer and owner to clarify and coordinate their respective visions and positions. Existing CASE-tools are oriented mostly to reflect a logical component of functional structure, not physical one dealing with parameters and evaluations. This paper propose a formal specification and graphical notation to represent not only logical, but also physical functional aspects.

Однією із структур, які можна виділити у розподіленій інформаційній системі (РІС), є функціональна структура (ФС). Вона визначає істотну для роботи системи якісну унікальність окремих компонент, можливість перерозподілу функцій та характеристики, які описують цю можливість. Завдання побудови та аналізу функціональної структури системи, переважно, вирішуються на етапі системного аналізу проектування РІС під час визначення та формалізації системи вимог замовника.

Для вирішення завдань автоматизації проектування РІС найживанішим інструментом сьогодні є CASE-системи [1,2]. Здебільшого, для опису функціональної структури із застосуванням CASE використовують потокові діаграми (DFD) та діаграми ієрархії функцій (FHD). Вони дають змогу провести декомпозицію головної функції системи на структуру функцій (мережу функцій). У сучасних CASE системах DFD та FHD широко застосовуються для специфікації вимог замовника, досягнення розуміння між замовником та виконавцем щодо принципів функціонування системи. Такі діаграми є основою для подальшої розробки інформаційної системи. Водночас CASE-системи обмежуються тільки логічним проектуванням, а DFD-діаграми досліджують логічні зв'язки між функціями. Модифікація таких діаграм у напрямку врахування параметричної компоненти функцій дає можливість вже на перших етапах проектування сформулювати та розв'язати багато задач, зібрати необхідні початкові дані та побудувати структури, необхідні для подальшого фізичного проектування системи. Власне на цих етапах можна

сформулювати та отримати орієнтовні оцінки параметрів системи з прив'язуванням до функціональної діаграми (яку розуміє замовник і може дати коректні оцінки).

Розвиток сучасних інформаційних систем також вимагає використання побудованої ФС не тільки на етапі проектування, але протягом всього життєвого циклу системи для застосування в інтелектуальних інформаційних мережах. Це ставить додаткові вимоги до засобів подання, відображення, моделювання та супроводу ФС ПІС.

Для використання у фізичному проектуванні визначимо окремі сутності модифікованої потокової діаграми.

Клієнт — це інформаційний процес, що звертається з запитам до сервісів системи. Клієнт може бути визначений як

$$Cl = \{idCl, PbCl, M(Zp), CmCl\},$$

де $idCl$ — ідентифікатор клієнта, $PbCl$ — блок параметрів, $M(Zp)$ множина запитів, $CmCl$ — коментар.

У множині запитів кожний запит, крім змістовної частини, має ще й вектор параметрів та коментар.

$$Zp = \{idZp, SnZp, PbZp, CmZp\},$$

де $idZp$ — ідентифікатор запиту, $SnZp$ — зміст запиту, $PbZp$ — блок параметрів, $CmZp$ — коментар. До параметрів запиту, наприклад, можуть належати обмеження щодо часу відповіді, інтенсивність генерування запитів, об'єми інформації, які в середньому передаються під час запиту, допустимі формати відповіді та ін. З погляду сервісу системи клієнт повністю визначається його множиною запитів.

Сервіс — це сутність, яка обслуговує запити клієнтів. Сервіс може виступити клієнтом інших сервісів, якщо цього буде вимагати обслуговування запиту. Модель сервісу має складнішу структуру. Сервіс доцільно поділити на інтерфейсну частину та реалізаційну частину (рис. 1).

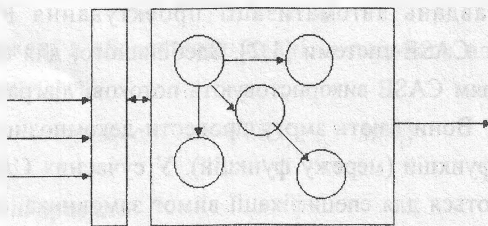


Рис. 1. Структура сервісу

Інтерфейсна частина сервісу — це частина сервісу, яку клієнт бачить. З точки зору клієнта можна вважати, що інтерфейс сервісу — це і є сервіс. Інтерфейс є набором певних викликів з визначеними параметрами і достатньо фіксований. Звертання до інтерфейсної частини, наприклад, може виконуватися єдиною мовою звертання до ресурсів мережі. Фіксованість та стандартизованість інтерфейсів дає змогу:

- одночасно підтримувати різні реалізаційні частини моделі сервісу,

- довізначати ці частини,
- одночасно підтримувати як наочні, графічні описи-моделі, так і програмно-автоматні реалізаційні блоки різного рівня складності; всі моделі одного сервісу ізоморфні, існують правила відображення моделей однієї в іншу;
- коректно опрацьовувати ситуацію, коли опис моделі відсутній або неповний.

З іншого боку опис системи як сукупності сервісів є зручним, надає системі гнучкості, можливості багаторазового використання компонент, стійкості до змін.

Сервіс визначається як

$$Se = \{idSe, inSe, reSe, CmSe\},$$

де $idSe$ — ідентифікатор сервісу, $inSe$ — інтерфейсна частина, $reSe$ — реалізаційна частина, $CmSe$ — коментар.

На етапі побудови функціональної діаграми доцільно абстрагуватися від деталей реалізації, а описувати сервіс тільки його інтерфейсом.

Інтерфейс сервісу —

$$inSe = \{idSe, M(ISe)\},$$

де $idSe$ — назва сервісу, $M(ISe)$ — множина функцій, точок входу запитів.

Сховище (інформаційне або матеріалів). Це традиційний елемент потокової діаграми. Використовується для вирішення завдань визначення структур даних.

Ресурсу відповідає якийсь матеріальний або технічний об'єкт. Наприклад, мережа, конференцзал, обладнання для презентації. Важливим параметром ресурсу є його "пропускна здатність" — опис параметрів одночасного обслуговування запитів користувачів. Доступ до сховищ даних та ресурсів подається так само, як і звертання до сервісу. Але крім суто функціональної, "поведінкової" компоненти, сховища та ресурси мають додаткові компоненти (структури даних або модель обслуговування запитів згідно з особливостями реального об'єкта). Ресурс можна розуміти як предмет або іншу сутність, необхідну під час роботи сервісу. Якщо при звертанні одного сервісу до іншого перший сервіс очікує відповіді другого, то при звертанні до ресурсу відбувається резервування частини пропускної здатності ресурсу, але сервіс продовжує працювати.

Формально ресурс можна подати як

$$Re = \{idRe, PbRe, MdRe, CmRe\},$$

де $idRe$ — ідентифікатор ресурсу, $PbRe$ — блок параметрів ресурсу, $MdRe$ — параметрична модель, $CmRe$ — коментар.

Параметрична модель ресурсу — це певна формула або інша модель, яка дозволяє визначити для наявної завантаженості ресурсу час закінчення обслуговування запиту з певними вимогами (параметрами) щодо завантаження цього ресурсу.

На етапі проектування функціональної структури системи доцільно визначити типових користувачів системи, що виступають її клієнтами та характеризуються однотипною професійною діяльністю. Для таких типових клієнтів визначаються відповідні ролі. Прикладом такої ролі є роль працівника бухгалтерії, розробника

програмного забезпечення, адміністратора ресурсів мережі та ін. Аналогічний підхід може бути застосовано і до інших типів клієнтів, наприклад, організацій або застосувань.

Визначення типових ролей дає змогу:

- а) зменшити розмірності задачі аналізу та проектування;
- б) отримати дані для подальшої розробки та побудови профілів клієнтів, забезпечення якості обслуговування кожної категорії клієнтів,
- в) налагодити мережу на потреби категорій клієнтів, визначити параметри обслуговуючих сервісів, проектувати мережі під потреби клієнтів,
- г) отримати інформацію для подальшої розробки схеми прав доступу до інформаційних ресурсів та сервісів мережі.

Типізація клієнтів може проводитися

- апіорі при проектуванні системи на підставі аналізу технології роботи ролі або експертних оцінок;
- апостеріорі, в результаті аналізу діяльності клієнтів наявної системи для уточнення параметрів функціонування мережі та профілів клієнтів. Переважно, такі оцінки будуються на основі статистичних даних, зібраних за достатньо великий проміжок часу.

Подальший аналіз проводиться для кожної ролі окремо. Визначається схема звертання до сервісів мережі. Вона будується на основі аналізу потреб професійної діяльності ролі. Визначається множина операцій ролі, числові параметри кожної операції. Наприклад, такими параметрами можуть бути частота виконання операції, часові обмеження щодо отримання відповіді, об'єм інформації, який передається та отримується в результаті операції. Кожна така елементарна операція — це взаємодія з певним сервісом мережі або “внутрішня” операція (без взаємодії з сервісом). Таким чином аналізується технологічний процес певної ролі, будується схема технологічного процесу (процесів) з оцінкою параметрів проміжних операцій. Якщо неможлива чи недоцільна побудова та аналіз схем технологічного процесу, зв'язки між роллю та сервісом можуть бути визначені експертно. Визначені операції та їх параметри фіксуються, наприклад, у таблиці такого вигляду:

	Сервіс	Параметри					Коментар
		1	2	3	...	n	
Клієнт 1	Операція 1						
	Операція 2						
	...						
	Операція К						
Клієнт 2							

Параметри кожної операції доцільно поділити на групи згідно з розмірністю параметра. Приклади найвживаніших параметрів:

- θ — частота виконання операції,
- $\lambda_{вх}$ — об'єм інформації, що передається в запиті до сервісу (клієнт → сервіс);
- $\lambda_{вих}$ — об'єм інформації, що передається у відповідь на запит (сервіс → клієнт),
- $\tau_{від}$ — обмеження щодо часу відповіді,
- π — пріоритет (важливість) операції.

При початковому аналізі зв'язків між клієнтами та сервісами можна визначити

- дозволені та недозволені операції;
- основні та другорядні операції;
- часті та епізодичні операції.

Після побудови схем технологічних процесів вони узагальнюються у схему звертання до сервісів мережі. Спочатку роль та сервіс з'єднує максимум один зв'язок (асоціація) з відповідним вектором параметрів.

$$LCl = \{idLCl, idCl, idSe, PbLCl, CmLCl\},$$

де $idLCl$ — ідентифікатор зв'язку, $idCl$, $idSe$ — ідентифікатори клієнта та сервіса, що зв'язані, $PbLCl$ — блок параметрів зв'язку, $CmLCl$ — коментар.

Якщо зв'язок узагальнює декілька операцій, то параметри кожного типу визначаються за параметрами складових операцій за правилами, окремими для кожного типу параметрів. Наприклад, для наведених вище параметрів можуть бути використані такі правила:

$$\theta' = \sum \theta_i \quad \lambda'_{вх} = \sum \lambda_{вх} \quad \lambda'_{вих} = \sum \lambda_{вих}$$

$$t'_{від} = \min(t_{від}) \quad \pi' = \max(\pi)$$

Зв'язки між клієнтами та сервісами всієї системи можна відобразити та проаналізувати у матричних діаграмах з використанням інструменту 'Matrix diagrammer'.

	Сервіс1	Сервіс2	СервісМ
Клієнт 1			
Клієнт 2			
Клієнт I		зв'язок I2	
Клієнт K			

Спочатку, після визначення основних типів запитів та їх параметрів функціональна діаграма подана одним контекстним сервісом (так, як у DFD) (рис. 2). Контекстний сервіс, як і контекстна діаграма, специфікує зв'язки контекстного сервісу з зовнішніми

сутностями (клієнтами). На відміну від контекстної діаграми зовнішні сутності тут не подані, що підкреслює можливість обслуговування запитів визначених типів та форматів незалежно від їх джерела. Отже, головним елементом контекстної діаграми є специфікація запитів.

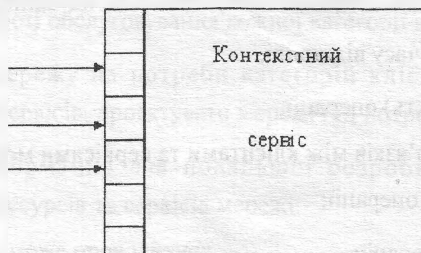


Рис. 2. Контекстний сервіс

Далі у процесі проектування розв'язується задача декомпозиції контекстного сервісу у мережу сервісів.

Це дає змогу:

- глибше зрозуміти структуру виконання запитів клієнтів;
- отримати дані, необхідні для подальшого проектування;
- сформувані дані для розв'язання аналітичних параметричних задач над мережею сервісів.

Процес декомпозиції сервісу — творчий, неформальний. Загалом, при такій декомпозиції можна керуватися двома правилами:

- кожний сервіс орієнтується на виконання закінченого набору функцій.
- сервіс визначається так, щоб його можна було використовувати багатократно, для обслуговування як “зовнішніх клієнтів”, так і інших сервісів, що виступають у ролі клієнтів.

Можна сказати, що сервіс мусить бути спеціалізованим на виконання визначеного вузького кола споріднених функцій і одночасно універсальним в діапазоні вхідних параметрів запитів (максимально широкий діапазон допустимих параметрів для кожної спеціалізованої функції сервісу).

Сервіс може бути деталізований з використанням множини діаграм виконання запитів до сервісу. Кожний вид запиту з інтерфейсу сервісу ілюструється окремою діаграмою. Ця діаграма подає процес виконання запиту, логічну послідовність звертання до сервісів більшого рівня деталізації.

Зауважимо, що діаграма виконання сервісу ілюструє послідовність звертання до окремих сервісів. Неявно на ній присутній один координуючий процес, який формує запити до відповідних сервісів, отримує та аналізує відповіді, приймає рішення. На цій

діаграмі не відображаються міжсервісні запити. Виняток становлять запити до ресурсів та сховищ даних. Міжсервісні запити переносяться у діаграми деталізації відповідних сервісів. Це спрощує результуючу діаграму.

Основні елементи діаграми виконання сервісу такі.

- Сервіс (рис.3).

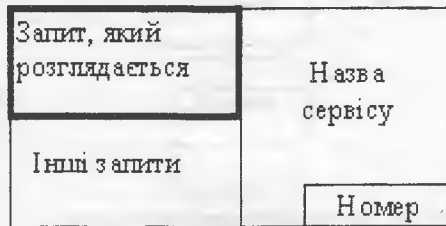


Рис. 3. Діаграма сервісу

Номер сервісу ідентифікує сервіс у діаграмі.

- Операція.(рис.4).

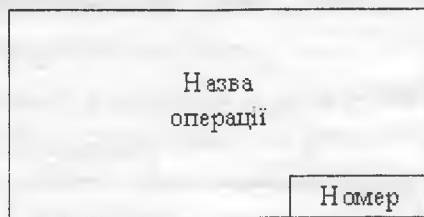


Рис. 4. Діаграма операції

Операція, на відміну від сервісу, не призначена для обслуговування багатьох клієнтів. Вона описує певний набір внутрішніх для цього сервісу дій, спрямованих на виконання запиту. Операція локальна для свого сервісу. Для подання параметричної компоненти операції та порядку її декомпозиції використовуються ті самі підходи, що й для опису сервісу.

- Зв'язок (потік) (рис.5).

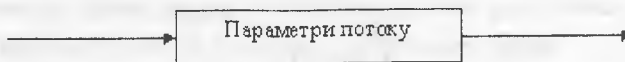


Рис. 5. Діаграма потоку

Серед параметрів потоку (ПП) вказуються оцінкові параметри, аналогічні до параметрів зв'язку між роллю та сервісом. Параметри зв'язку можуть описувати числові характеристики як запиту до сервісу, так і відповіді сервісу.

- Розгалуження потоку.

а) умовне розгалуження потоку (рис.6.).

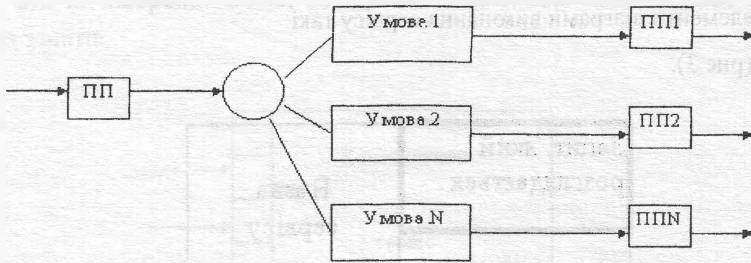


Рис. 6. Умовне розгалуження потоку

Умови 1-N задаються на множині параметрів потоку та повинні в сукупності повністю покривати весь діапазон параметрів вхідного потоку. Параметри ПП1, ПП2, ..., ППN — можуть бути підмножною вихідних параметрів ПП.

Крім умовної вершини розгалуження, можлива також, як варіант, ймовірнісна вершина, що вказує на розгалуження, окремі гілки яких відбуваються з певними ймовірностями p_i (рис.7.)

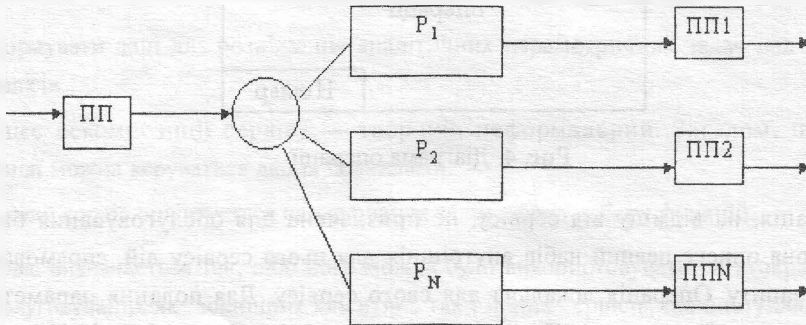


Рис. 7. Ймовірнісне розгалуження потоку

$$\sum_{i=1}^N p_i = 1$$

Ймовірнісні розгалуження особливо корисні, коли параметри потоку невідомі та оцінюються експертно з великим ступенем неточності.

· Сховище даних.

Для подання сховища даних використаємо загальноживані позначення.(рис.8)

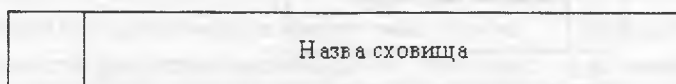


Рис. 8. Діаграма сховища даних

· Ресурс (рис.9).

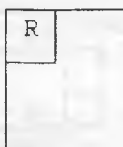


Рис. 9. Діаграма ресурсу

Правила, які задовольняє коректна діаграма, такі.

- У діаграмі є один вхідний потік та вихідні потоки, які відповідають запиту, що розглядається.
- Діаграма параметрично коректна, тобто описує обслуговування запиту для всього можливого діапазону параметрів запиту.
- Забороняється рекурсія у деталізації сервісів — сервіс не може звертатися до самого себе прямо або непрямо, через низку сервісів- посередників.

Іншою діаграмою, що описує сервіс, є діаграма функціональної ієрархії — перелік сервісів та операцій, до яких реалізується звертання під час виконання запиту до сервісу (рис. 10). Ця діаграма є ієрархічним деревом. Коренем цього дерева є контекстний сервіс. На кожному рівні дерева перераховано операції та сервіси, які наявні у діаграмі декомпозиції сервісу вищого рівня.

Така діаграма подібна до популярної діаграми ієрархії функцій (Function Hierarchy Diagram), але, на відміну від неї, окремі сервіси у ній можуть повторюватися на різних рівнях діаграми.

Діаграма ієрархії звертань до сервісів дозволяє перевірити умову відсутності рекурсивного звертання сервісу до самого себе (у кожному шляху на діаграмі від довільної вершини до кореня дерева кожний сервіс може зустрічатися тільки один раз.) Кожна операція може зустрічатися тільки один раз у всьому дереві.

Діаграма ієрархії звертань до сервісів дозволяє визначити *рівень сервісу* — міру загальності, абстрактності сервісу. Рівнем сервісу будемо вважати номер рівня ієрархії, на якому сервіс зустрічається вперше. Контекстний сервіс має рівень 0.

Якщо сервіс не деталізується далі, назовемо його атомарним сервісом.

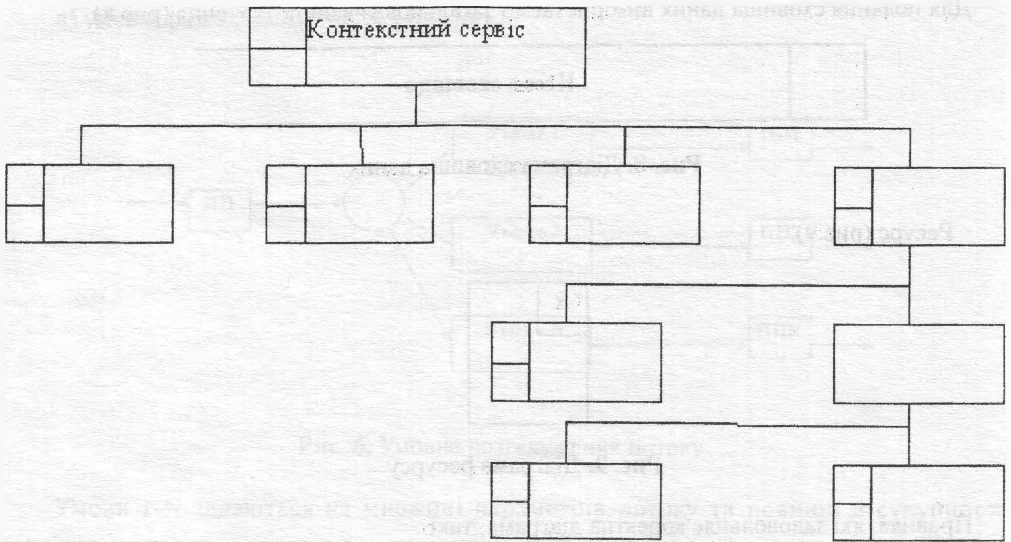


Рис. 10. Діаграма функціональної ієрархії

Над діаграмами сервісів можна визначити низку операцій аналізу та перетворення, які корисні у процесі проектування.

- Декомпозиція сервісу Se

У результаті декомпозиції сервісу Se_i отримують множину діаграм $\{Dt_i\}$, кожна з яких відповідає одному з інтерфейсних запитів сервісу $IfSe_i \in M(IfSe)$.

- Об'єднання сервісів. $Se = Se_i \cup Se_j$

У результаті об'єднання двох сервісів Se_i та Se_j отримують сервіс Se , в якому множина інтерфейсів є об'єднанням множин їх інтерфейсів.

$$M(IfSe) = M(IfSe_i) \cup M(IfSe_j)$$

Однакові запити зливаються в один. Діаграми виконання сервісів також об'єднуються.

- Перетин сервісів. $Z = Se_i \cap Se_j$ — це множина однакових, взаємозамінних запитів та діаграм їх виконання.

- Взаємозалежність сервісів $Zal(Se_i, Se_j)$

Взаємозалежність визначається як частота сумісного виклику кількох сервісів $f(Se_i, Se_j)$ у відношенні до загальної кількості викликів цих сервісів.

$$Zal(Se_i, Se_j) = f(Se_i, Se_j) / (f(Se_i) + f(Se_j))$$

Вимірюється на множині запитів Zp певної ролі Cl за визначений період t .

- Ядро сервісів.

Ядро сервісів — це сукупність сервісів $Se' \subset Se$, які викликаються найчастіше.

$$f(Se') = \max(f(Se))$$

При цьому розглядаються тільки “атомарні” сервіси. Визначення ядра сервісів може бути корисне для підвищення продуктивності всієї системи, оскільки для виконання ядра можна використати продуктивніше обладнання. Визначається на множині запитів Zp певної ролі Cl за визначений період t .

Інтенсивність обміну сервісу Se також визначається на множині запитів певної ролі Cl за визначений період t .

Виділяється

- зовнішня інтенсивність обміну — це сума об’ємів інформації, які сервіс отримує у запити та видає у відповіді.

$$\Lambda_z = \sum \lambda_{вх} + \sum \lambda_{вих}$$

- внутрішня інтенсивність обміну $\Lambda_{вн}$ — це сума об’ємів інформації, які передаються між сервісами з урахуванням послідовної декомпозиції запиту.

$$\Lambda_{вн} = \sum \lambda_{вн}$$

де $\lambda_{вн}$ — інтенсивність обміну внутрішнього потоку сервісу

- загальна інтенсивність обміну. Сума зовнішньої та внутрішньої інтенсивності.

$$\Lambda_{за} = \Lambda_{вн} + \Lambda_z$$

На підставі побудованої функціональної структури можна сформулювати та розв’язати ряд задач, корисних як для проектувальника, так і для адміністратора системи, зокрема:

- оцінювання часу виконання запиту;
- визначення ступеня завантаженості сервісів множиною запитів клієнтів;
- визначення ядра сервісів для різних категорій клієнтів.

Розв’язок цих задач при різних значеннях параметрів дозволить спрогнозувати параметри поведінки системи.

Розробка, аналіз та діаграмне відображення ФС на етапі системного аналізу визначає вхідні дані для опису та моделювання поведінки сервісів під час параметричного проектування на наступних етапах розробки РІС.

1. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. <http://www.citforum.ru>, 1998. 2. Калянов Г.П. CASE. Структурный системный анализ. М.1996.