

УДК 519.9

Зубенко В.В.<sup>1</sup>, к.ф.-м.н., доц.

## Про комунікативну інформатику

*Розглядається комунікативна платформа інформатики.*

*Ключові слова:* комунікативний процес, комунікативний система, інформатика.

<sup>1</sup> Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глущкова 4д, е-mail: [vvz@unicyb.kiev.ua](mailto:vvz@unicyb.kiev.ua)

V.V.Zubenko<sup>1</sup>, doc.

## On the communicative informatics

*The communicative platformot of informatics is desided.*

*Key Words:* communicative process, informatics, communicative system.

<sup>1</sup> Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d, e-mail: [vvz@unicyb.kiev.u](mailto:vvz@unicyb.kiev.u)

Статтю представив д.ф.-м.н., проф. Буй Д.Б. (за результатами конференції ТАAPSD'2012)

Не зважаючи на стрімкий розвиток індустрії інформатики протягом останніх кількох десятиріч, процес її становлення як фундаментальної науки все ще не можна вважати завершеним. Перебігає він досить суперечливо між кількома напрямами – інженерним, математичним, комунікативним, а останнім часом і інформаційним. Кожен із них має своє підґрунття й зорієнтований на ті чи інші аспекти процесів обробки інформації. Такий плюралізм у підходах до визначення задач та предмета інформатики актуалізує проблему пошуку її базової парадигми (парадигм) та ідентифікації як наукової дисципліни.

Робота присвячена комунікативній платформі інформатики та уточненню предмета інформатики у цій платформі.

1. Традиційно, коли говорять про інформатику, то мають на увазі сферу збереження, пошуку, передачі та обробки інформації [1] і ті проблеми, які тут виникають (інженерні, наукові тощо). Важливо, що при цьому не можна обмежитися інформацією взагалі – потрібна певна її структуризація. А саме, коли мова йде про інформаційні об'єкти, то в них явно виділяють повідомлення, його значення та зв'язок між ними. Перші є формою, завдяки якій інформацію ідентифікують і передають, другі відповідають за зміст інформації і враховуються при її перетворенні, а треті (комунікативні) подають спосіб зв'язку між першими двома, та,

моживо, з іншими інформаційними об'єктами<sup>1</sup>. Слід зазначити, що в теорії програмування – важливій складовій інформатики - елемент зв'язку довгий час, якщо й не був поза кадром, то без належної уваги. З ускладненням людино-машинних систем та появую об'єктно-орієнтованого та інших напрямків сучасного програмування ситуація кардинально змінюється і цей комунікативний елемент стає важливою складовою теоретичних досліджень програм.

У світлі становлення науки інформатики було б цікаво подивитися як формувалися сусідні, вже класичні, точні науки (а саме такою ми хочемо бачити й інформатику). Якщо взяти, наприклад, фізику, то побачимо, що: 1) вона як природнича наука має справу з певними моделями природних явищ (руху, твердого тіла, рідини, звуку, світла тощо), 2) ці моделі є фізичними за суттю, математичними за формулою, 3) їх може бути не одна, а декілька, навіть для одного і того самого явища; наприклад, корпускулярна та хвильова теорія світла тощо, 4) кожна з моделей, орієнтуючись на відображення тих чи інших аспектів природних явищ, спирається на певну методологічну платформу, 5) моделі не є чимось сталим, а час від часу в процесі

<sup>1</sup> Є спроби трактувати інформацію більш загально та долучати до інформатики аспекти, не пов'язані безпосередньо з передачою та обробкою інформації [2]. На наш погляд, вони мають стосуватись вже не інформатики, а якоїсь іншої науки, наприклад, такої як Information Science в США і в Західній Європі чи інформології в РФ тощо.

розвитку удосконалюються і збагачуються новими фактами (в межах платформи), б) самі платформи теж можуть змінюватись і, як наслідок, можуть з'являтися нові, якісно інші, моделі тих самих явищ.

Як відомо (див., наприклад, [3].), фізика розпочиналась з вивчення механіки та оптики на базі механістичної платформи Ньютона, потім перейшли до теорії поля (електрика) і нарешті до квантової фізики (теорія відносності). Нові і старі платформи можуть якийсь час співіснувати паралельно, доповнювати одна одну (корпускулярна і хвильова теорія світла), відходити у минуле з передаванням своїх здобутків наступникам (корпускулярні елементи у квантовій платформі) та відмирати з часом (як планетарна модель Пталомея).

Повертаючись до інформатики, можемо зробити висновок, що плюралізм серед існуючих в ній моделей та платформ явище цілком природне. Питання полягає в іншому - у цілях та якості існуючих моделей, на скільки вони адекватні поставленим задачам тощо.

На сьогодні однією з найбільш загальних та вивчених платформ інформатики є платформа, відома під назвою *дискретні системи* [4]. Завдяки своєму апріорі високому рівню абстрактності вона знайшла широке коло застосувань в інформатиці, але в багатьох випадках цей рівень є занадто високим і потребує суттєвого зниження. Серед інших платформ, насамперед, виділимо загальні та спеціалізовані платформи програмування. Однією з найбільш загальних та глибоких є *сумісна* платформа [5], яка є розвитком більш ранніх платформ – *деноаттивної* [6,7] та *композиційної* [8-10,11,12], направлених на уточнення та вивчення семантики програм. Спеціалізовані платформи, відомі під назвами *парадигм програмування*, орієнтуються на побудову програм і підтримують різних стилів програмування, кожний з яких характеризується своїми моделями програм і даних, засобами їхнього опису й методами реалізації.

Нас цікавлять комунікативні аспекти процесів обміну та обробки інформації, і розробка математичної платформи для моделювання таких аспектів.

У комунікативних системах [13] обміну та обробки інформації (КС)<sup>2</sup> обов'язковою є наявність:

- 1) предметної області (ПрО) з певною сукупністю інформаційних об'єктів та співвідношень між ними;
- 2) суб'єкта-ініціатора, що формує і передає суб'єкту-обробнику певну вхідну та приймає від нього певну вихідну інформацію;
- 3) суб'єкта-обробника, що приймає вхідну інформацію, аналізує її, обробляє та повертає як вихідну суб'єкту-ініціатору.

Далі суб'єкти-ініціатори та суб'єкти-обробники будемо називати скорочено *ініціаторами* та *обробниками*. Головне призначення комунікативних систем полягає у здійсненні комунікативних процесів, кожний з яких розгортається в певному часовому просторі і встановлює зв'язок між певними об'єктами предметної області системи. За своюю роллю в процесі ці об'єкти розподіляються на *вхідні* та *вихідні*. З перших розпочинається процес, іншими закінчується. Комунікативний процес складається з етапів, які разом утворюють його *життєвий цикл*.

Останній розпочинається ініціатором, який формує й передає обробнику за допомогою засобів зв'язку повідомлення, що містить *запит*<sup>3</sup> на обробку певних вхідних інформаційних об'єктів. Запит окрім останніх містить інформацію про *мету* обробки, яка може бути сформульована неявно – як вимоги до очікуваних вихідних об'єктів або явно – як детальний опис їх отримання за вхідними об'єктами. Перший спосіб превалює, наприклад, в пошукових інформаційних системах, другий – в обчислювальних системах з процедурними мовами програмування. В обох випадках мета декларує певне співвідношення між об'єктами і сама подається у вигляді спеціального інформаційного об'єкту – *програми*. Вихідні дані є результатом

<sup>2</sup> У кібернетиці подібні системи називають системами зі зворотним зв'язком.

<sup>3</sup> Не змішувати з запитами в інформаційно-пошуковими системах. Тут термін “запит” трактується як загальне поняття з більш широким змістом.

реалізації програми (мети) на конкретних вхідних даних.

Запит оформлюється у вигляді набору дескрипцій<sup>4</sup> і сам є спеціальним інформаційним об'єктом, зрозумілим обробнику. Останній приймає запит, виконує в межах певного часового інтервалу відповідні дії для реалізації програми й повертає ініціатору за допомогою засобів зв'язку отримані й декодовані вихідні дані.

Таким чином, кожний запит породжує певне співвідношення між об'єктами ПрО. Це співвідношення може бути не кінцевим, а тільки проміжним. У деяких випадках ініціатор повинен уміти аналізувати вихідні дані. Після відповідного аналізу (або без нього) він формує новий запит, знову передає його обробнику і так далі аж доки не буде отримано кінцевий результат комунікативного (діалогового) процесу.

Зазвичай обробник має власні інформаційні об'єкти – *внутрішні* (сукупності яких утворюють його стани), на відміну від *зовнішніх* - тих, що фігурують в предметній області. Тому сам запит містить не вхідні об'єкти обробника, а тільки їхні прообрази, які після кодування утворюють початковий стан обробника. Це стосується й вихідних даних, поданих у заключному стані процесу обробки. Вони потребують декодування. В деяких системах з метою безпеки вхідні й вихідні дані перед кодуванням (декодуванням) шифрують (дешифрують).

Ключовим моментом в комутативному процесі є його реалізаційний компонент – частина процесу, яка відбувається “на території” обробника. Отримавши запит, обробник трансформує його у вхідний стан та в послідовність дій, які необхідно виконати, щоб перейти у заключний стан. В теорії цю послідовність дій називають *обчисленням*. В кожний момент часу обробник перебуває у певному стані, в якому подана та чи інша інформація. Обчислення зводиться до цілеспрямованої зміни станів обробника. Щоб змінити стан, обробник виконує певну дію (операцію). Існує механізм, який визначає порядок застосування операцій в обчисленні. Є два варіанти реалізації цього механізму. Перший - характерний для *спеціалізованих* систем, які

орієнтовані на обробку запитів з фіксованою метою. У цьому випадку порядок застосування операцій відображеній безпосередньо у структурі обробника, він є її елементом. Другий притаманний, так званим, *універсальним* системам, які можуть обробляти запити з різною метою. Базується він на використанні обробником спеціальних інформаційних об'єктів – *внутрішніх процедур*, які задають правило для проведення обчислень - визначають над якими об'єктами і які дії необхідно виконати на кожному з кроків обчислениння. В структурі такого обробника є механізм для перевода (трансляції) програм запитів у внутрішні процедури (або сукупності процедур), які їх реалізують. У цьому випадку процес обчислениння складається з двох фаз – трансляції та виконання внутрішньої процедури на вхідних даних.

На практиці зустрічаються і системи з мішаним варіантом реалізації запитів, коли одні запити реалізуються як спеціалізовані, а інші – за допомогою універсального механізму.

Комуникативні системи теж мають свій життєвий цикл, пов'язаний з їх появою та функціонуванням у часовому просторі. Він складається з етапів і розпочинається з аналізу варіантів вхідних даних та усіх можливих запитів системи. Наступні етапи охоплюють: проектування і побудову обробника та експлуатацію системи.

В межах комунікативної платформи інформатику можна було б визначити як *науку, що вивчає моделі комунікативних процесів та систем*. Однак, зважаючи на дискусійний характер обговорення, зробимо кілька принципових зауважень щодо даного означення.

Насамперед зазначимо, що в такому вигляді воно постає занадто загальним і багатоаспектним. Моделі комунікативних процесів і систем, які є спрощеним поданням реальних комунікативних процесів та систем, можуть дуже відрізнятися й вимагати принципово різних підходів. Достатньо постатися на різноманіття тільки кібернетичних моделей (від систем автоматичного регулювання, ймовірнісних і статистичних моделей до автоматних та комп’ютерних моделей). Тому було б доречно обмежити моделі комунікативних процесів і систем.

<sup>4</sup> Дескрипція (від англ. description – опис) – мовна конструкція, що замінює назву предмета, виражаючи його зміст іншими мовними засобами.

Опустимо далі розгляд спеціалізованих систем, де вирішальну роль відіграє специфіка запитів та її відображення в структурі обробника, і зосередимо увагу на **універсальних** системах. В залежності від способу, яким обробники знаходять вихідні дані в таких системах, вони поділяються на *породжуючі* та *i синтезуючі*. У першому випадку обробник породжує певну надмножину вихідних даних (або використовує вже готову таку надмножину як, наприклад, у пошукових системах), з якої відбирає останні, згідно з умовами, які визначено у внутрішній процедурі як інформаційному об'єкті, у другому – вихідні дані генеруються (будуються) з вихідних даних за допомогою цілеспрямованих імперативних дій. Обидва способи широко представлені в комунікативних системах.

Далі, нагадаємо, що інформація в комунікативній системі представляється інформаційними об'єктами. В кожній з них є свої правила побудови таких об'єктів. Без цих правил і без розуміння їх суб'єктами системи ніякий комунікативний процес неможливий. Модель комунікативної системи повинна враховувати цей аспект, а це значить, що вона має бути **дескриптивною**. Дескриптивність моделі означає, що до складу її елементів входять дескриптивні системи (ДС), в яких подаються (описуються) усі її інформаційні об'єкти: вихідні і вихідні дані, запити та внутрішні процедури. Мова йде не про одну, а, як правило, про декілька ДС, тому що інформаційні об'єкти різних суб'єктів системи потребують різних дескрипторологічних засобів. Як мінімум – *мову специфікації* запитів для ініціатора і *внутрішню мову* для обробника. Остання описує внутрішні інформаційні об'єкти та процедури. Ці мови можуть співпадати, але тільки тоді, коли обидва суб'єкти КС співпадають.

Таким чином, в центрі уваги комунікативної платформи мають бути не просто моделі комунікативних процесів і систем, а саме дескриптивні їхні моделі.

Характеристичною особливістю таких моделей є те, що в них присутній часовий фактор і властивості деяких об'єктів (їхні значення і навіть їхній тип та зв'язки) можуть з часом змінюватись. Про такі об'єкти говорять, що вони є мінливі (*динамічні*). Моделі, які допускають динамічні об'єктами, самі називаються *динамічними*. Це означає,

що моделі комунікативних систем і процесів апріорі мають бути **динамічними**.

Відомо три види дескриптивних моделей – екстенсіональні, інтенсіональні й мішані (інтегровані) в залежності від того, якими є їхні ДС.

Для **екстенсіональних** ДС характерним є предикативний опис об'єктів через їх обсяг. Вони виділяють об'єкти за конкретним іменем *v* і певними спільними ознаками його можливих значень або за тією роллю, яку вони відіграють у тих чи інших контекстах. Подібні ознаки та ролі отримали назву - *характеристичних властивостей*. Характеристичні властивості описують об'єкти таким чином, що можливі розбіжності між ними поза межами цих властивостей не приймаються до уваги. Одну й ту ж роль можуть грати абсолютно різні за значенням об'єкти. При екстенсіональному підході здійснюється перехід на вищий рівень абстракції до узагальненого об'єкта - *змінної*, яка має своє ім'я - *v*, а значенням - клас допустимих значень (екстенсіонал).

Можна конкретизувати змінну, пов'язавши з її ім'ям *v* конкретне допустиме значення *a*. Факт конкретизації змінної подається тріадою  $v \mapsto a$ . Конкретне ім'я *v* абстрактне у тому сенсі, що воно нікак не пов'язане з конкретною структурою його поточного значення, яка міняється від значення до значення. Слід зазначити, що предикат може описувати і рівно один об'єкт (і його можна навіть взяти за його ім'я!), але і в цьому випадку він не буде мати прямого відношення до структури значення. Наприклад, лінійне рівняння  $2x = 8$  може розглядатися як синонім для свого єдиного розв'язку (числа 4), але воно немає ніякого відношення до структури числа 4 в десятковій системі числення.

В **інтенсіональних** моделях кожний з об'єктів представлений не тільки конкретним своїм ім'ям, але і своїм конкретним значенням (*інтенсіоналом*). При цьому, на відміну від екстенсіональних моделей, структури імен та значень, як правило, певним чином пов'язані, а саме так, що за іменем може бути установлено його значення ( в екстенсіональних моделях для цього потрібен ще додатковий компонент-зв'язок ).

Різницю між екстенсіональними та інтенсіональними засобами проілюструємо наступними двома важливими прикладами.

Перший стосується натуральних чисел. З одного боку, їх можна визначити інтенсіонально, за допомогою числа 0 і відповідної конкретної сукупності операцій збільшення на 1 або скориставшись якоюсь позиційною, наприклад, звичайною десятковою, системою числення, взявши відповідну конкретну кількість одиниць, десятків, сотень тощо. В обох випадках значення числа – є результат конкретних дій в часі, які обумовлені його іменем. З іншого боку існує екстенсіональне означення натурального числа як кардинального через потужність множин.

Якщо взяти, наприклад, число три. Інтенсіонально його можна подати виразами  $1+1+1$  чи десятковою цифрою 3. Екстенсіонально ж воно визначається як кількість елементів (кардинал) в таких множинах як { груша, яблуко, слива }, { a,b,c } та усіх їм рівнопотужних. Характеристичною властивістю тут є предикат штибу  $tri(x)$  = “множина  $x$  - рівнопотужна множині { груша, яблуко, слива }”. Кількість елементів в цих множинах – це те, що об'єднує усі триелементні множини. Ім'я кардинала “три” замінює їх в усіх контекстах, пов'язаних з їхньою потужністю<sup>5</sup>.

Другий приклад пов'язаний з поняттям функції. Відомо два підходи до означення функції: процедурний – з функціями як конкретними правилами для обчислення результату (інтенсіональний) та теоретико-множинний – з функціями як відношеннями у теорії множин (екстенсіональний).

*Мішані* моделі використовують обидва види засобів. Моделі обробників навіть конкретних комунікативних систем є **мішаними**. Це випливає з подвійного характеру внутрішніх процедур, кожна з яких має, і інтенсіонал - правило для знаходження вихідних даних за вхідними, і екстенсіонал - сукупність обчислень, яку це правило породжує. Наявність екстенсіоналу (він може бути і одноелементним) є наслідком масовості внутрішніх процедур. Масовість означає здатність процедури застосовуватись до вхідних даних з певної не порожньої їх сукупності (немає жодного варіанту вхідних даних – немає і жодного обчислення!). Це

вимагає, щоб останні були представлені в процедурі екстенсіонально - не конкретними значеннями (хоча і це можливо у випадку одного єдиного варіанту вхідних даних), а своєю роллю в обчисленні.

І нарешті про головне. Якщо звернутись до реальних обробників КС, то побачимо, що їхні ДС є **фінітними**, тобто в них усі об'єкти, і зовнішні, і внутрішні є скінченно поданими. Це стосується, як запитів, так і внутрішніх процедур. КС та їхні моделі з фінітними ДС називаються **конструктивними**.

Наголосимо окремо, що конструктивність КС розповсюджується не на всю предметну область КС, а тільки на ту її частину, яка представлена у мові специфікацій КС. В останній представлена (наблизено або точно) певна конструктивна модель предметної області. На цьому базується ідея інформаційного моделювання в різних галузях.

Враховуючи сказане, запропонуємо таке загальне визначення предмета інформатики в комунікативній платформі:

**Комунікативна інформатика** – це наука, що вивчає конструктивні моделі комунікативних процесів і систем.

Моделі, що вивчає комунікативна інформатика, будемо назвати **комунікативними інформаційними**, як і самі конструктивні комунікативні процеси та системи.

**2. Конструктивний характер комунікативних аспектів інформаційних моделей** фіксує граници комунікативної інформатики та вberігає її від надмірного узагальнення й ототожнення з іншими науками, насамперед з кібернетикою як наукою про загальні закони перетворення інформації в системах зі зворотним зв'язком та з інформологією – узагальнюючою наукою про інформацію в цілому, усі її вияви в живій і неживій природі, властивості й види загальних інформаційних процесів, а також з інформаційною наукою (Information Science). Взаємовідносини між цими науками, з одного боку, та комунікативною інформатикою – з іншого, приблизно такі, як між загальним поняттям і та його конкретизацією.

Як приклад можна навести взаємовідносини між загальною теорією груп і групою дійсних матриць або між класичним математичним аналізом і конструктивним

<sup>5</sup> Але воно не може це зробити в тих же теоретико-множинних операціях, кулінарних рецептах тощо, де суттєвими є конкретні елементи множин.

тощо. Останній приклад є особливо характерним: класичний аналіз базується на теоретико-множинній (екстенсіональній) платформі, а конструктивний - на алгоритмічній (інтенсіональній). І як математичні теорії вони дуже різні, хоч і мають значний спільний перетин (спільні теореми тощо). Якщо порівняти доведення однакових теорем в обох теоріях, то в конструктивному аналізі вони, як правило, значно довші і складніші. Про велику різницю між конструктивними і неконструктивними методами свідчать і складність отримання точних оцінок поведінки функцій чи процесів (конструктивних за означенням) у порівнянні з аналогічними асимптотичними (екстенсіональними) оцінками в межах навіть однієї теорії.

Таким чином, не можна ототожнювати кібернетику чи інформаційну науку з комунікативною інформатикою та стверджувати, як іноді можна сьогодні

почути, що, наприклад, кібернетика вже "вмерла", залишилася лише інформатика тощо. Це просто різні, хоч і споріднені науки зі своїми предметами та методами дослідження. Це ж стосується і таких сучасних напрямів інформаційної науки як соціальна інформатика, біоінформатика тощо, в яких комунікативні інформаційні моделі є лише засобом дослідженъ, а предмети в них різні.

**Висновки.** В роботі введено поняття комунікативної інформатики як науки, що вивчає конструктивні моделі комунікативних процесів і систем. Комунікативна платформа інформатики є доволі загальною, але хочеться сподіватися, і достатньо змістовою для того, щоб відобразити основні проблеми теорії й практики обробки та передачі інформації, у тому числі, і програмування та сприяти їхньому розв'язку.

1. *Le Garf A. Dictionnaire de l'informaticque.* Paris: Presses Universitaires de France. 1982.
2. *Kolin K. K. Evolution of Informatics.* /K.K. Kolin // Information technologies.- 2005, № 1, c.2-16
3. *Einstein A, Infeld L. The growth of ideas from early concepts to relativity and quanta.* – NY: Simon & Shuster, 1954.
4. *Kapitonova J.V., Letichevskij A.A. Mathematical theory of computer systems design.* – M.: Nauka, 1988.- 296 c.
5. *Red'ko V.N. The descriptive foundations of the essential platform / V.N. Red'ko, I.V.Red'ko, N.V.Grishko // Problems of Programming – 2008. – № 2-3. – P. 75-83.*
6. *Scott D.S. Outline of a mathematical theory of computation.*- Technical Monograph RPG-3, Programming Research Group, Uni. of Oxford, 1970.
7. *Stoy, J. E. The Scott-Sterachey Approach to Programming Language Theory / J. E. Stoy.* – Cambridge, Massachusetts: The MIT Press, 1979.
8. *Basarab I.A., Nikitchenko M.S., Red'ko V.N. Compositional databases.* – K.: Lybid', 1992. – 192 p.
9. *Red'ko V.N., Brona J., Buj D.B., Poljakov V.A. Relational databases: Table algebras & SQL-like languages.* -K.: Publishers «Akademperiodika», 2001.-198 p
10. *Nikitchenko M.S. Theory of programming. Part I / M.C. Нікітченко – Nizhin: Publishers of Mykoly Gogolja NDU, 2010.- 119 p.*
11. *Red'ko V.N. Foundations of compositional programming. / V.N. Red'ko // Programming. – M., 1979. – № 3.*
12. *Zubenko V. Applied Program Algebra.* – Kiel: CAU Kiel.-1985.- Bericht. – № 8504.- 50 p.
13. *Zubenko V.V., Omel'chuk L.L. Programming.* – K.: Publishers of University of Kyiv, 2011. - 625 .c.

Надійшла до редколегії 03.12.2012