

Національна академія наук України
Інститут телекомунікацій і глобального інформаційного простору

Захарченко Тарас Леонідович



УДК 004.021

**КОМПОЗИТОСУТНІСНІ МОДЕЛІ АДАПТИВНИХ
ПРОЦЕСОНАЛЬНИХ СЕРЕДОВИЩ**

05.13.06 – Інформаційні технології

Автореферат
дисертації на здобуття наукового ступеня
кандидата технічних наук

Київ – 2018

Дисертацією є рукопис.

Робота виконана в Національному технічному університеті України “Київський політехнічний інститут імені Ігоря Сікорського” Міністерства освіти і науки України.

Науковий керівник:

доктор фізико-математичних наук, професор
Редько Ігор Володимирович,
Національний технічний університет
України “Київський політехнічний інститут
імені Ігоря Сікорського”, професор кафедри
конструювання електронно-обчислювальної
апаратури.

Офіційні опоненти:

доктор технічних наук, старший науковий
співробітник

Стрижак Олександр Євгенійович,
Національний центр «Мала академія наук
України», м.Київ, заступник директора з
наукової роботи

кандидат технічних наук, доцент

Олецький Олексій Віталійович,
Національний університет «Києво-
Могилянська академія»,
доцент кафедри мультимедійних систем

Захист відбудеться “05” липня 2018 р. о 13⁰⁰ годині на засіданні спеціалізованої вченої ради Д26.255.01 Інституту телекомунікацій і глобального інформаційного простору НАН України за адресою: 03186, Київ 186, Чоколівський бульвар, 13, ауд. 601.

З дисертацією можна ознайомитись у бібліотеці Інституту телекомунікацій і глобального інформаційного простору НАН України за адресою: 03186, Київ 186, Чоколівський бульвар, 13.

Автореферат розісланий “04” червня 2018 р.

Учений секретар
спеціалізованої вченої ради, к.т.н.



О.Г. Лебідь

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Будь-яка людська діяльність на кожному етапі її розвитку обумовлена її інтуїтивним баченням або розумінням. Природно, що найбільш спрощене розуміння за необхідністю присутнє на перших етапах розвитку діяльності. Але поступово, з розвитком самої діяльності, еволюціонує і її розуміння. Поштовхом для наступного кроку еволюції зазвичай виступає притаманна діяльності проблемність – надмірне спрощення її розуміння. Не винятком тут є й інформаційно-технологічна діяльність (ІТ-діяльність, ІТД), пов'язана з індукованими нею інформаційно-технологічними процесами (ІТП).

Причин проблемності загалом більш ніж достатньо, але, що стосується ІТД, то тут головною серед них бачиться надто спрощене, а значить – недостатньо змістовне розуміння взаємодоповнення відповідних процесів та їх результатів. Загальна природа взаємодоповнюваності дуже складна та розкриття її в повному обсязі є, напевно, неможливим. Реальним бачиться шлях поступового прагматико-обумовленого збагачення уявлень про нього.

Домінуючу роль у цьому взаємодоповненні традиційно відіграють результати ІТД. Для цього більш ніж достатньо об'єктивних підстав. Добре відомі вражаючі результати, досягнуті, насамперед, у математиці, які, без сумніву, є наслідком подібного домінування. Але безпрецедентно широке поширення комп'ютерної справи в усіх областях дійсності не могло не прийти в суперечність із реаліями розвитку всіх сфер людської діяльності. Змістовно кажучи, прийняте домінування результатів ІТД як позиція чітко підтвердило, що абсолютизація будь-якого, навіть виключно важливого і продуктивного позиціонування, часто призводить до вкрай негативних результатів. Адже абсолютизація будь-чого в силу її природи замкнена в актуальних даностях предмету абсолютизації. Це свідчить про нездатність адекватно реагувати на обумовлені потенційною відкритістю об'єктивних процесів пізнання еволюційні зміни бачення дійсності. Звідси виходить, що усунення головної причини недоліків у рамках актуально замкненої точки зору на ІТД є неможливим. Актуально замкнене загальнозначуще ядро розглядів як їх логіка повинно бути адекватно збагачене потенційно відкритим різноманіттям його предметних продовжень. Тобто традиційні сьогодні, актуально замкнені у своїй відкритості або замкненості ІТ-парадигми мають бути якісно збагачені, відповідно, до їх взаємодоповнення – відкрито-замкненої парадигми. Змістовно така зміна бачиться у переході в ІТД від продукування окремих, іноді досить ефективних, але фактично обмежених домінантою результату ІТД підходів до вирішення задач, до розробки єдиної відкрито-замкненої ІТ-платформи, домінантою якої є не результат ІТД, а ІТД як діяльність, що націлена на процеси вирішення ІТ-задач. Це дозволяє не протиставляти існуючі точки зору на ІТ-діяльність, а навпаки, природним чином об'єднувати їх у якості тих чи інших предметних продовжень «спільного знаменника» – взаємодоповнення ІТП та їх результатів, породжених в рамках ІТД. Така об'єднуюча точка зору складає кістяк реального розуміння ІТД. Вона дозволяє по-новому поглянути на

такий вид ІТД, як побудова апаратного та програмного забезпечень та доповнити його точними дослідженнями та розробками.

У дисертації в якості такого «спільного знаменника» різних точок зору на ІТД пропонується концепція адаптивного процесонального середовища (АПС). Основу його складає забезпечення гнучкості ІТД шляхом адекватного зміщення акцентів розгляду з виключно результатів вирішення задач, зокрема їх верифікації на коректні методи їх досягнення. Це забезпечує можливість ставити та ефективно вирішувати задачі управління *якістю* отримуваних рішень, забезпечення *ефективності* діяльності з пошуку таких рішень і проблему збереження *інвестицій* у такій діяльності. Тут під якістю розуміється коректність продукованих АПС рішень, їх гарантоздатність та функціональна безпека. Під ефективністю ж розуміється підвищення продуктивності вирішення ІТ-задач за рахунок простоти використання існуючих суміжних рішень і уточнення самого ІТП за допомогою концепції АПС, прийняття рішень в умовах часткової невизначеності. Отримати рішення цих та інших задач стало можливим завдяки залученню до розглядів, у першу чергу, процесів їх вирішення, зокрема адекватній організації цих процесів. Останнє забезпечується реальним врахуванням активної участі суб'єкта в ІТП, там, де ця участь дійсно необхідна. Із зазначеного безпосередньо виходить, що створення згаданого адаптивного процесонального середовища є **актуальною та важливою** задачею.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота відповідає напрямкам досліджень кафедри конструювання електронно-обчислювальної апаратури НТУУ «КПІ», а її результати отримані в рамках науково-дослідної роботи «Прискорення обчислень з використанням пристроїв, що реконфігуруються» (РК №0113U001874) згідно з основними науковими напрямками діяльності Національного технічного університету України «Київський політехнічний інститут» та пріоритетного напрямку розвитку науки і техніки України «Інформаційні та комунікаційні технології».

Мета і завдання дослідження. Метою дисертаційної роботи є підвищення якості та ефективності ІТД шляхом розробки методу вирішення задач, що продукує композитосутнісні моделі та взаємодоповнює існуючі точки зору на ІТД й враховуючи як традиційні, так і прагматико-обумовлені, нетрадиційні аспекти ІТД. З урахуванням основних вимог до композитосутнісних моделей АПС і у відповідності до зазначеної мети в дисертації ставляться і вирішуються наступні задачі:

- розробка та обґрунтування концепції адаптивного вирішення задач та створення на її базі відкрито-замкнених засад АПС;
- побудова на базі відкрито-замкнених засад АПС логіко-предметної понятійної системи АПС та створення на цій основі композиційних засад АПС;
- створення на основі композиційних засад АПС композитосутнісної понятійної системи, основу якої складає поняття композитосутнісного відношення;

- прагматико-обумовлена типізація композицій і розробка методу композитосутнісної релятивізації;
- дослідження семантичних характеристик репрезентативних класів обчислювальних функцій над зліченими носіями;
- розробка та аналіз дослідної реалізації АПС, створеного на базі композитосутнісної моделі АПС.

Об'єкт дослідження дисертаційної роботи – інформаційно-технологічні процеси вирішення задач.

Предмет дослідження дисертаційної роботи – композитосутнісні моделі АПС у застосуванні до процесів розробки апаратного та програмного забезпечень.

Методи дослідження – проведені в дисертаційній роботі дослідження базуються на сукупності загальнометодологічних, логіко-епістемологічних, логіко-математичних методів. Серед методологічних основними є *метод сутесутнісної релятивізації* і його композитосутнісна спеціалізація, серед логіко-епістемологічних – уведення та виключення абстракції. До використаних логіко-математичних методів відноситься композиційний метод вирішення задач й алгебраїчні методи дослідження. Також використано семантичний аналіз.

Наукова новизна одержаних результатів, що виноситься на захист, полягає в наступному:

- вперше створено композиційні та композитосутнісні засади, композитосутнісну понятійну систему АПС і доведено теорему про нерухому точку проєктивної функції, що дозволило сформулювати як теоретичний фундамент АПС, так і композитосутнісний метод розробки апаратного та/або програмного забезпечень, що складає основу адаптивного вирішення задач на практиці.
- вперше встановлено прагматико-обумовлену типізацію композицій, що змістовно збагачує ключове в роботі поняття композиції;
- отримав подальший розвиток метод отримання алгебраїчних характеристик класів обчислюваних функцій, що забезпечило можливість дослідження АПС, зокрема вирішення проблеми повноти в репрезентативних класах обчислювальних функцій;
- вперше виявлені семантико-синтаксичні структури HDL-мов, що дозволило визначити їх структурно-функціональну модель і скласти єдиний фундамент для розробки в рамках АПС апаратного забезпечення.

Таким чином, у дисертаційній роботі вперше висунута та розроблена концепція композитосутнісних моделей АПС, яка дозволяє розробляти засоби вирішення ІТ-задач, що адекватно відображають природу цих задач і є теоретичною основою для практичної імплементації АПС.

Практичне значення одержаних результатів визначається зокрема:

- розробленою дослідною реалізацією АПС, яка підтримує наскрізне інтерактивне вирішення задач, починаючи з властивої їм прагматики та

закінчуючи автоматичною генерацією коду рішення в формальних мовах Verilog HDL та C, що дозволило дослідити практичне застосування АПС;

- виділеними практично важливими класами функцій над записами та вирішеними проблемами їх повноти в примітивних програмних алгебрах;
- створеними графічною та текстовою мовами специфікації композицій і вирішеним із їх використанням набором репрезентативних задач, що дало змогу продемонструвати ефективність вирішення ІТ-задач в АПС.

Основним і найзагальнішим практичним результатом є розроблена композитосутнісна модель адаптивного процесонального середовища, що дозволяє коректно ставити та вирішувати проблеми управління якістю рішень і забезпечення ефективності розробки програм.

Одержані в дисертації нові результати знайшли застосування в організації ТОВ «Відео Інтернет Технології» (м. Київ), використані при виконанні НДР НТУУ «КПІ», а також у навчальному процесі кафедри конструювання електронно-обчислювальної апаратури (КЕОА) факультету електроніки НТУУ «КПІ», що підтверджується відповідними актами:

– акт від 04.02.2016 про використання в ТОВ «Відео Інтернет Технології» дослідної реалізації АПС, орієнтованої на розробку програмного забезпечення для обробки зображень;

– акт від 02.12.2016 про використання в результатах НДР (РК №0113U001874) дослідної реалізації АПС для побудови на її базі надійних і ефективних засобів розробки апаратного забезпечення;

– акт від 02.12.2016 про впровадження в навчальний процес кафедри КЕОА факультету електроніки НТУУ «КПІ» дослідної реалізації АПС в курсах “Основи побудови інформаційно-обчислювальних засобів інтеграції”, “Експертні системи” на практичних роботах студентів, пов’язаних із програмними алгебрами.

Особистий внесок здобувача. Основні ідеї та наукові результати дисертаційної роботи отримані автором особисто та висвітлені у десяти працях, шість з яких виконані одноосібно [1] [2] [3] [4] [5] [6].

У роботах, опублікованих у співавторстві, здобувачу належать:

- [7] – формування базового набору частково-рекурсивних функцій і предикатів над множиною записів, теорема про те, що цей набір складає породжуючу систему ППА;
- [8] – визначення I_m^n -базису чр-функцій і чр-предикатів над записами, побудова відображень елементів множини мультимножин на елементи множини записів, теорема про те, що цей базис складає допустиму систему ППА;
- [9] – визначення I_m^n -базису чр-функцій та чр-предикатів над записами, що зберігають денотати, побудова відображень елементів множини векторів на елементи множини записів, теорема про те, що цей набір складає породжуючу систему ППА;
- [10] – Розділи 1-5, 7, 6.2.

Апробація роботи. Наукові та практичні результати дисертаційної роботи доповідались і обговорювались на міжнародних та вітчизняних науково-технічних семінарах і конференціях:

- Всеукраїнській науково–практичній конференції «Innovations in science and technology 2012»;
- IEEE 34th International Conference on Electronics and Nanotechnology (ELNANO 2014);
- Міжнародній конференції молодих вчених «Електроніка-2012»;
- Міжнародній конференції молодих вчених «Електроніка-2013»;
- XIII Всеукраїнській науково-практичній конференції студентів, аспірантів і молодих вчених «Теоретичні та прикладні проблеми фізики, інформатики та математики».

Публікації. За темою дисертації опубліковано **10** наукових праць: 5 в українських фахових виданнях, кожна з яких індексується щонайменше в двох міжнародних наукометричних базах даних – РИНЦ і EBSCO [1] [3] [6] [7] [8] [9]; одна стаття в працях міжнародних наукових конференцій, включених у наукометричну базу даних IEEEExplore Digital Library [2]; а також дві тези доповідей на міжнародній і всеукраїнській науково-технічних конференціях [4] та [5]; патент України на корисну модель № 72313 «Швидкодіючий багатосаровий перцептрон, що гнучко масштабується» [6]; монографія «Концептологічні основи проектування» [10].

Структура та обсяг роботи. Дисертаційна робота складається з переліку умовних позначень, вступу, чотирьох розділів, висновків, списку використаних джерел загальний обсяг дисертації становить 164 стор.

Основний зміст роботи

У **вступі** обґрунтована актуальність роботи, визначена мета, задачі, об'єкт і методи дослідження дисертації. Розкрита новизна основних наукових та практичних результатів, отриманих здобувачем та їх місце в розглядуваному напрямі теоретико-практичних досліджень.

Перший розділ «**Загальнозмістовні передумови адаптивних процесональних середовищ**» присвячений огляду традиційних підходів до вирішення ІТ-задач, їхній конструктивній критиці та обґрунтуванню актуальності розвитку гнучких (адаптивних) технологій розробки. Суть його, у першому наближенні, полягає в тому, щоби реально, а не тільки номінально залучати під час вирішення ІТ-задач наряду з їх рішеннями також і відповідні процеси, причому не еkleктично, а в контексті властивого їм причинно-наслідкового зв'язку. У своїй основі запропонований підхід спирається на розвинутий Редьком В.Н. композиційний підхід.

На загальному рівні будь-який інформаційно-технологічний процес розглядається в контексті адекватного взаємодоповнення основних аспектів вирішення задач. Перший із них, як заведено говорити, синтаксичний, пов'язаний із правильністю побудови форм, із внутрішньою їхньою структурою і т.п. Другий аспект, його називають семантичним, стосується самої суті рішень

задач, їхньої природи та структури, відношень у множині смислів або значень, представлення «складних» сутностей через більш прості й т.д. Останній аспект, іменованій прагматичним, зачіпає комплекс залежностей між тими, хто створює рішення та їх виконавцями, а також самими рішеннями. Такий багатоаспектний взаємодоповнюючий розгляд ІТП обумовлений її потенційною відкритістю, передусім, в тому, що стосується прагматики та прагматико-семантичної взаємодії у розробці рішень.

З цієї позиції розглянуто низку традиційних ІТ-платформ, зокрема: функціональна, об'єктно-орієнтована, логічна, структурна, модульна, денотаційна. За результатами розглядів сформульовано низку положень про те, що всі вони є замкнутими у властивих їм актуальностях – чи то функція, чи то модуль, чи то об'єкт, штучно обмежена сукупність генетичних структур, тощо. Із цього, враховуючи потенційну відкритість ІТД та індукованих нею процесів, безпосередньо впливає висновок про неможливість у рамках традиційних підходів адекватно вирішувати задачі залучення ІТП і їх наслідків у розгляді. Цим обґрунтовується необхідність розвитку нового підходу до ІТД.

Розділ «**Теоретичні засади адаптивних процесональних середовищ**» присвячено формуванню концепції адаптивних процесональних середовищ (АПС) та її обґрунтуванню. Підґрунтям тут є розвинута в роботах Редька І.В. теорія дескриптивних середовищ та застосування її результатів саме до ІТ-задач. Основу будь-якої, зокрема і ІТ-діяльності як мотивованого процесу досягнення мети, становить її розуміння. При цьому, розуміння ІТ-процесів має релятивну (сутесутнісну) природу. Така релятивність обумовлена, передусім, визначальним у ІТ-діяльності значенням активної ролі суб'єкта в ній. Тому вона є необхідним атрибутом розуміння ІТ-процесів, що підтримує його природну здатність до еволюції. У цьому контексті, ІТ-процес розглядається як діяльність, можливий наслідок якої обумовлено відповідною передумовою, тобто, у термінах теорії дескриптивних середовищ, як кондитологічна діяльність. Кондитологічна парадигма ІТП становить змістовний фундамент роботи. Усі подальші розгляди фактично підпорядковані прагматико-обумовленим, як безпосереднім, так і опосередкованим збагаченням цієї парадигми.

В основі будь-якої кондитологічної діяльності лежить так зване сутесутнісне відношення (ССВ) – відношення між сутністю, що досліджується (обумовлюється) – предметом діяльності, та наслідком її обумовлення – відповідною суттю. Таким чином, приходимо до сутесутнісної парадигми ІТД: ІТД експлікативно зводиться до розкриття суті її предмету. Ключове значення ССВ полягає в тому, що саме воно є носієм властивої ІТД релятивності й тому реально враховує активну роль суб'єкта в ІТП. Індуковане **ІТ-діяльністю** сутесутнісне відношення, очевидно, є рефлексивним, антисиметричним і транзитивним, тобто є частковим порядком. Це, у свою чергу, дозволило збагатити сутесутнісну парадигму ІТД відповідним сутесутнісним положенням: ІТД – суть рефлексивно-транзитивне замикання сутесутнісного відношення. Дане положення збагачує введenu вище експлікацію ІТД та реально підтримує

собою індуковані ІТД інформаційно-технологічні процеси генезису рішень у еволюційному розвитку розуміння ІТД.

Зі сказаного безпосередньо випливає, що змістовну суть ССВ становить обумовлення – зв'язування умовами, обставинами і т.п. Тому подальші збагачення уявлень про ССВ, а значить і про ІТ-діяльність пов'язані зі збагаченням уявлень саме про індуковані ІТ-діяльністю обумовлення та зі створенням адекватних, тобто релятивних, а значить таких, що враховують активну роль суб'єкта в ІТП, передумов для цього. Відкрито-замкнена парадигма ІТД [10] вбачається тут адекватною платформою. Вона дозволяє поглянути на обумовлення через призму динаміко-статичної взаємодії процесу та його наслідку. Причому процес є домінантою розгляду.

Основним атрибутом обумовлення, як виду роду діяльності, є його суб'єктно-об'єктна природа. Остання ж тісно пов'язана з активною й пасивною формами її здійснення. Змістовно кажучи, активна форма є прерогативою суб'єкта обумовлення й пов'язана з відповідним ІТП (активно впливає на нього), а пасивна – пов'язана, як із наслідком ІТП, так, очевидно, і з об'єктом обумовлення. Взаємодоповнення цих форм ІТП є ключовою характеристичною особливістю породжуваного сутесутнісного відношення. Тому подальше збагачення ССВ здійснимо завдяки відносній автономізації активної та пасивної складових обумовлення. Це дає змогу здійснити ще одне реальне і принципове збагачення ІТП. А саме, прагматико-обумовлено виділити такі види сутностей, як концепт і монаду та залучити в розгляди новий активно-пасивний різновид ССВ – концептно-монадне відношення (КМВ). А відповідно, збагатити ІТП принципово новим видом обумовлення – концептуванням. Принциповість концептно-монадної взаємодії у ІТП полягає, передусім, у тому, що цей вид обумовлення є носієм таких засадничих особливостей ІТП, як релятивність, зокрема, врахування активної ролі суб'єкта, відкрито-замкненість та динаміко-статичність. А відтак, КМВ є принциповим збагаченням ССВ та становить загальнозначущий фундамент ІТ-діяльності. Отже, у контексті прагматики ІТП маємо:

Теза зведення. ССВ експлікативно зводиться до КМВ.

КМВ суттєво збагачують сутесутнісну парадигму ІТД. Для подальших загальнозначущих збагачень поки що недостатньо передумов. Тому всі подальші збагачення носять явно виражений предметний характер. У роботі проведена низка таких принципових збагачень, що дало змогу цілком обґрунтовано, спираючись, зокрема, на принципи функціональності, композиційності та процесійності залучити до розглядів поняття композиції в ролі основного засобу для роз'яснення процесу побудови рішень. Водночас, композиціюванню притаманні всі ті засадничі властивості, про які згадувалось у контексті концептування. Це визначає шлях до нівелювання основних вад традиційних підходів до вирішення задач – домінування результату в ІТД, протиставлення підходів до вирішення задач один одному, нездатність адекватно враховувати активну роль суб'єкта в ІТ-діяльності та її вплив як на індуковані ІТП, так і на їх наслідки. Отже, композиціювання природно

розглядати як принципове предметне збагачення концептування в прагматиці ІТД, в тому числі розробки програмного та апаратного забезпечення. Основу цього збагачення становить поняття композитосутнісного відношення (КСВ) – обумовленої прагматикою даної роботи конкретизації сутесутнісного та концептномонадного відношень. Це поняття дало змогу принципово збагатити конкретикою принцип процесійності через розуміння ІТД як замикання КСВ. Останнє, з огляду на свої змістовні властивості, цілком природно розглядати як частковий порядок. Це дозволило залучити до розглядів КСВ традиційний математичний апарат і отримати результати, що не тільки збагатили, власне, КСВ, але й дали змогу суттєво просунутись у дослідженні загального поняття композиції.

Перехід до композиціонування в ІТ-задачах дав змогу не тільки нівелювати виявлені вади згаданих традиційних методів, а й інтегрувати останні з методами, що розвиваються в рамках денотативних підходів. Передумови широкого розвитку робіт у денотативному напрямі були закладені в роботах С. Кліні, А. Тарського, Д. Скотта, Ю.Л. Єршова, де Баккера та ін. До теперішнього часу на цьому шляху отримана велика кількість як теоретичних, так і прикладних результатів. По суті всі вони базуються на топологіях, індукованих тими чи іншими частковими порядками. У цьому розділі деякі результати денотаційного підходу були спроектовані на введені вище КСВ.

Рішення, індуковані реальними ІТП, складно влаштовані. Крім того, як впливає із дескриптивного аналогу теореми К. Гьоделя про неповноту, підвести під такі рішення змістовну концептуально єдину денотативну базу в принципі неможливо. Максимум на що можна тут розраховувати, це створити реальну основу для індивідуалізації в універсумі таких рішень їх типів, які б мали прагматико-обумовлені денотативні властивості.

Безпосередньо з вищенаведеної прагматико-обумовленої аргументації визначення композитосутнісних відношень виходить, що вони можуть розглядатися в якості такої основи. Принаймні вони дають змогу індивідуалізувати типи рішень (зокрема програмних чи апаратних), серед яких знаходять своє природне місце не тільки всі відомі нам типи традиційних ІТ-рішень (зокрема апаратні та програмні проекти), а і прагматико-обумовлені нові типи принципово більш багатих нетрадиційних рішень, індукованих реальним розумінням ІТД. В якості репрезентативного прикладу таких рішень може розглядатися будь-яка функція f , що задовольняє умову:

$f(\prod_{n=0}^{\infty} a_n) = \prod_{n=0}^{\infty} f(a_n)$, де $\prod_{n=0}^{\infty} a_n$ й $\prod_{n=0}^{\infty} f(a_n)$ – границі послідовностей сутностей, що

розуміються як найменші верхні грані в сенсі композитосутнісного часткового порядку між їхніми безпосередньо слідуючими сутностями. Такі функції в роботі називаються *проективними функціями*. Легко бачити, що це неперервні функції в топологіях, індукованих композитосутнісними відношеннями як частковими порядками. Тому через теорему Тарського про структуру множини нерухомих точок впливає, що має місце

Теорема. Проективні функції мають нерухомі точки.

Принципова значимість цієї теореми полягає, звичайно, не в доведенні існування нерухомих точок у проєктивних функцій, а в тому, що вона відкриває цілком природний шлях для дослідження тих типів прагматико-обумовлених рішень, для яких їх існування є більш-менш прямим наслідком такої обумовленості. Такий підхід є не просто відмінним від відомих традиційних, а й дуальним до них. Адже, розглядувані тут топології індуковані прагматико-обумовленими частковими порядками. Отримані результати складають логічну основу адаптивних процесональних середовищ. Усі наступні розділи дисертації за своєю суттю присвячені одному – наповненню цієї основи предметним змістом.

У першому наближенні адаптивне процесональне середовище (АПС) є інтеграційним середовищем, що поєднує, а не протиставляє різноманіття прагматико-обумовлених точок зору на предмети ІТД. Концептуально єдину основу розглядів становить поняття композитосутнісного відношення. Це середовище є результатом побудованої у роботі релятивізації універсального середовища інтеграції (УСІ), досліджуваного в роботах Редька І.В. ІТ-релятивізація УСІ представлена у роботі у вигляді як трирівневе взаємодоповнення низки категорій:

$$УСІ = \langle \langle \langle УР, УП \rangle, УЗП \rangle, ЗЕР \rangle,$$

де УР – універсум рішень в АПС, УП – універсум предметів чи композицій у пасивній ролі в АПС, УЗП – універсум засобів побудови чи композицій в активній ролі в АПС, ЗЕР – засоби еволюційного розвитку чи метакомпозиції в АПС. Таким чином, обґрунтовано висновок про те, що АПС являє собою УСІ, обмежене зазначеними початковими умовами. Зафіксуємо сказане у вигляді головної тези.

Головна теза. АПС експлікативно зводиться до ІТ-релятивізації УСІ.

Всі подальші розгляди будуть підпорядковані подальшому збагаченню цієї тези як на рівні прагматико-семантичних досліджень, так і побудов конкретних рішень, зокрема мова йде про розробку апаратних та програмних рішень. Так наведено більш деталізований опис процесу розробки з використанням АПС. У першому наближенні суть такого підходу до побудови апаратного та програмного забезпечень складається у відшукуванні адекватного компромісу між замкнутим та відкритим поглядами на них, який досягається завдяки прагматико-обумовленій, тобто такій, що враховує активну роль розробника, актуалізації вищезгаданих категорій, зокрема, УП та УЗП.

У ролі прикладу процесу розробки, до якого застосовні АПС, розглянута розробка реконфігурованих систем. Проведено критичний огляд сучасних підходів і обґрунтовано доцільність застосування розробки в АПС для цієї галузі.

У зв'язку з релятивізацією УСІ до ІТД, постає необхідність дослідження поняття композиції, яке є ключовим у розділі. Важливе значення тут мають такі загальні властивості композиції: *тотальність*, *адекватність* та *замкнутість*. Ці властивості підтримують прагматичну обумовленість і релятивність виокремлення композицій як засобів побудови прагматико-обумовлених рішень.

Змістовно кажучи, саме вони об'єктивізують визначальні для прагматики задачі відношення між ключовими суб'єктами рішення – його розробником і тим, на кого воно орієнтовано, тобто його користувачем. Так властивість *адекватності* забезпечує прагматичне обмеження композиції як алгебраїчної операції вимогою відповідності її змісту суб'єктивним уявленням про процес вирішення задачі. Властивість *замкнутості* також є релятивним обмеженням композиції вимогою забезпечити можливість сприйняття отриманого рішення його користувачем. Таке трактування є суттєвим узагальненням традиційного розуміння обчислюваності та природно зводиться до нього у разі, коли користувачем рішення є комп'ютер. Вимога ж *тотальності* фіксує той факт, що сьогодні немає прагматичних мотивацій розгляду не всюди визначених засобів генезису рішень.

Властивості тотальності, замкнутості та адекватності композицій суттєво збагачують їх як клас алгебраїчних операцій. Та все ж він усе ще є надто загальним, а значить - недостатньо змістовним. Необхідне його подальше збагачення. Головним інструментом тут є прагматико-обумовлена типізація (ТОП). За допомогою ТОП у цьому розділі здійснено низку важливих безпосередніх і опосередкованих збагачень поняття композиції, що покладені в основу прагматико-обумовленої класифікації всього класу композицій.

Одне з головних опосередкованих збагачень класу композицій, розглянуте тут, здійснено у спосіб прагматико-обумовленої типізації сутностей, на яких можуть бути задані композиції. Зазначена вище орієнтованість цієї роботи на інформаційно-технологічну проблематику, зокрема на розробку програмно-апаратних рішень, суттєво обмежує клас таких сутностей. За суттю, це звужує його до спеціальних класів функцій над носіями різних типів абстракції та рівнів загальності. Прагматико-обумовлена типізація цих класів є основним джерелом опосередкованих збагачень композицій.

Збагачення композицій природно одержувати шляхом ТОП класів функцій, на яких задані композиції, а збагачення в класі функцій визначати на основі ТОП структур даних. При цьому важливо підкреслити, що виходячи з вищезазначеного збагачення в класі функцій і даних мусять бути індуковані структурами композицій, а не навпаки. Що ж до властивостей функцій, то тут інтерес представляють ті з них, що використовуються композиціями. Одна з них фактично закладена у визначенні композиції як n -арної операції. Інша – у вимозі обчислюваності композицій і полягає в тому, що передбачається можливість знаходження значення будь-якої вихідної функції на будь-якому даному, якщо це значення визначене.

Суттєвий вплив на залучення тих або інших властивостей даних здійснює інформаційно-технологічна спрямованість роботи. Аналіз різних композицій для розробки апаратно та програмного забезпечень показує, що вони використовують властивості даних бути абстрактними даними, кортежами та множинами даних. Водночас можуть розглядатись кортежі та множини абстрактних, іменованих та метаіменованих елементів. Їх називають відповідно абстрактними, іменними та метаіменними кортежами та множинами.

З огляду на наведену ТОП даних та функцій, у роботі сформовано первинну класифікацію композицій, що включає абстрактні, іменні та метаіменні класи композицій. Щодо кортежного типу, то тут дослідження даних, функцій та композицій показало, що кортежність функцій є надто обтяжливою в контексті синтезу таких функцій. До того ж, кортежні структури можуть бути легко змодельовані, наприклад, іменними множинами.

У роботі проведено ґрунтовне дослідження класів абстрактних, іменних та метаіменних композицій, здійснена їхня подальша прагматико-обумовлена типізація. Зокрема розглянуто класи аплікативних і фундативних композицій як алгебраїчних операцій та їхні абстрактні, іменні та метаіменні підкласи. Отримані у рамках цього дослідження результати, безумовно, цікаві самі по собі. Однак у рамках цієї роботи вони важливі навіть не стільки самі по собі, скільки як такі, що забезпечили фактографічний фундамент для вирішення однієї з найважливіших задач роботи – дослідження загального поняття композиції.

Проведені вище дослідження носять поки що надто загальний характер. Для того, щоби мати можливість реалізовувати АПС на практиці та проводити семантичні дослідження, необхідно збагатити АПС, конкретизувавши її складові УР, УП та УЗП. У роботі це зроблено у вигляді відповідних композиційних систем, зокрема спеціальних видів алгебр.

У розділі «Семантичні дослідження АПС» проводяться його дослідження, зокрема, з використанням примітивних програмних алгебр (ППА) у застосуванні до розробки апаратного та програмного забезпечень. Мотивом вибору ППА була саме її простота. Це важливо, адже об'єктом дослідження є не самі композиційні алгебри, а їхні застосування в АПС. Звісно, якщо задачу можна вирішити в ППА, то це є своєрідною гарантією того, що вона може бути вирішена більш потужними засобами. Сигнатуру ППА складають композиції *суперпозиції, розгалуження та циклювання*, які являють собою адекватні уточнення основних методів конструювання програм, які властиві більшості високорівневих мов програмування.

У цьому розділі досліджено ППА на деяких важливих носіях, зокрема на класах функцій над записами, що можуть бути як маніпулятивними, що маніпулюють складовими елементами носія (у нашому випадку запису), не змінюючи їх, так і не маніпулятивними, які, в свою чергу, можуть змінювати складові елементи запису. Клас функцій над записами представляє великий практичний інтерес. Адже записи – популярна структура даних, яка лежить в основі key-value баз даних, хеш таблиць і т.д.

Під записом над множиною імен V та значень W (чи просто записом, якщо з контексту ясно, які V та W досліджують) будемо розуміти скінченне функціональне бінарне відношення між множинами імен V та значень W . Під денотатом розуміється елемент, який входить у структуру більш складного типу даних.

Для позначення записів домовимось користуватись прописними літерами I, J, K . Маленькими літерами u, v, w, \dots будемо позначати імена елементів

записів, літерами a, b, c, d, \dots – їхні значення, а літерами $\lambda, \mu, \eta, \dots$ – власне, елементи записів (денотати). У всіх випадках у разі потреби можна використовувати індекси.

Коротко розглянемо наступну сукупність маніпулятивних $Z^{(N, N)}$ -чр-функцій та $Z^{(N, N)}$ -чр-предикатів на записах, акцентуючи увагу на їх змістовних трактовках: $\sigma_{Z^{(N, N)}, \beta} = \{=Z, Od, Od^{-1}, Zn, \nabla, \div^Z, Mn, Rn, Nm^+, \{(1, 0)\}, I_m^n\}_{m=1, \dots, n}$.

Функція стандартизації імен Rn перетворює будь-який запис у запис зі стандартними іменами у відповідності з початковою впорядкованістю елементів вихідного запису:

$$Rn(\{(v_1, d_1), \dots, (v_n, d_n)\}) = \{(1, d_1), (2, d_2), \dots, (n, d_n)\} \Big|_{v_1 < v_2 < \dots < v_n}, \quad Rn(\{(v, d)\}) = \{(1, d)\}, \\ Rn(\emptyset) = \emptyset.$$

Функція кодування денотату Od ставить у відповідність будь-якому непорожньому запису новий запис, котрий є «одичним кодом» елемента запису з найменшим іменем. Порожній запис перетворюється сам у себе, інакше кажучи, $Od(\{(v_1, d_1), \dots, (v_n, d_n)\}) \Big|_{v_1 < v_2 < \dots < v_n} = \{(1, 1), (2, 1), \dots, (d_1 + 1, 1)\}$, $Od(\emptyset) = \emptyset$.

Функція декодування денотату Od^{-1} . Змістовно ця функція є оберненою до тільки що приведеної Od . Формально:

$$Od^{-1}(L, \underbrace{\{(v_1, 1), (v_2, 1), \dots, (v_k, 1)\}}_{\text{те, що декодується}}, \{(v_{k+1}, d_1), \dots, (v_n, d_n)\}) = \underbrace{\{(v_1, k-1), (v_{k+1}, d_1), \dots, (v_n, d_n)\}}_{\text{результат декодування}}, \quad \text{де } L -$$

деякий запис, котрий містить денотат $k-1$, $v_1 < v_2 < \dots < v_k < \dots < v_n$, $d_1 \in N \setminus \{1\}$.

Функція кодування імені Zn . Змістовно ця функція аналогічна введеної вище функції Od , але кодується вже ім'я, а не денотат. Строго кажучи: $Zn(\{(v_1, d_1), \dots, (v_n, d_n)\}) \Big|_{v_1 < v_2 < \dots < v_n} = \{(1, 0), (2, 0), \dots, (v_1 + 1, 0)\}$, $Zn(\emptyset) = \emptyset$.

Функція видалення за зразком \div^Z : $\div^Z(I, J) = pr_{pr_1(I) \setminus pr_1(J)}(I)$. Інакше кажучи, ця функція видаляє з першого аргументу всі елементи, імена яких містяться у другому аргументі – зразку. Для частинного випадку $pr_1(I) \cap pr_1(J) = \emptyset$, $\div^Z(I, J) = I$. Тут pr_i – функція проєкції за i -ю компонентою ($1 \leq i \leq m$) m -арного відношення, а $pr_{\{v_{i_1}, \dots, v_{i_k}\}}$ – допоміжна параметрична функція проєкції запису $pr_{\{v_{i_1}, \dots, v_{i_k}\}}$, котра будь-якому запису, наприклад $I = \{(v_1, d_1), \dots, (v_n, d_n)\}$, ставить у відповідність новий запис $pr_{\{v_{i_1}, \dots, v_{i_k}\}}(I) \equiv \{(v_{j_1}, d_{j_1}), \dots, (v_{j_p}, d_{j_p})\}$, де $\{v_{j_1}, \dots, v_{j_p}\} \equiv \{v_{i_1}, \dots, v_{i_k}\} \cap \{v_1, \dots, v_n\}$ та $pr_{\{v_{i_1}, \dots, v_{i_k}\}}(I) \subseteq I$.

Вибір елемента з мінімальним іменем Mn :

$Mn(\{(v_1, d_1), \dots, (v_n, d_n)\}) \Big|_{v_1 < v_2 < \dots < v_n} = \{(v_1, d_1)\}$. У випадку з пустим записом функція повертає пустий запис \emptyset : $Mn(\emptyset) = \emptyset$.

Функція слідування імен Nm^+ . Збільшує імена елементів непустиого запису на одиницю: $Nm^+((v_1, d_1), \dots, (v_n, d_n)) = \{(v_1 + 1, d_1), (v_2 + 1, d_2), \dots, (v_n + 1, d_n)\}$. Для пустиого запису: $Nm^+(\emptyset) = \emptyset$.

Функція накладання ∇ . Для будь-яких $I, J \in Z^{(N,N)}$ $\nabla(I, J) = J \cup pr_{pr_1(I) \setminus pr_1(J)}(I)$.

Функція ініціалізації $\{(1,0)\}$ ставить у відповідність будь-якому аргументу запис $\{(1,0)\}$.

Предикат рівності $=_Z$ вводитьься як звуження на множину $Z^{(N,N)}$ стандартного предиката рівності множин.

В подальших побудовах доведена

Теорема. $\sigma_{Z^{(N,N)}, \beta}$ – система утворюючих алгебри $A_{Z^{(N,N)}, \beta}^{cp}$, де алгебра A -- примітивна програмна алгебра.

Далі розглянемо такі функції та предикати на множині записів, які не зберігають денотати.

Предикат рівності $=_Z$ такий же, як і для маніпулятивних функцій.

Включення в запис $\overset{+}{U} : \overset{+}{U}(I) \equiv \begin{cases} I \cup \{\max(pr_1(I)) + 1, 0\}, \text{ якщо } I \neq \emptyset \\ \{(0,0)\}, \text{ якщо } I = \emptyset \end{cases}$, для

всіх $I \in Z^{(V,W)}$.

Вибір за максимальним іменем \max : $\max(I) \equiv pr_{\max(pr_1(I))}(I)$;

Обнуління значень $\{0\}$: $\{0\}(I) \equiv J$, де $pr_1(J) = pr_1(I) \& pr_2(J) = \{0\}$.

Інкремент \uparrow : ставить у відповідність до будь-якого «непустиого» запису $I \in Z^{(V,W)}$ запис $\uparrow(I)$ такий, що $\uparrow(I) = \{(v, a + 1) \mid \forall (v, a) \in I\}$;

Декремент \downarrow : ставить у відповідність будь-якому «непустому» запису $I \in Z^{(V,W)}$ запис $\downarrow(I)$ такий, що $\downarrow(I) = \{(v, b) \mid \forall (v, a) \in I \& b = \begin{cases} a - 1, a > 0 \\ 0, a = 0 \end{cases}\}$. У

випадку, якщо $I = \emptyset$, то і $\uparrow(I) = \downarrow(I) = \emptyset$.

Позначимо $\sigma_{Z^{(N,N)}} = \left\{ =_Z, \nabla, \overset{+}{U}, \max, \{0\}, \uparrow, \downarrow, I_m^n \right\}_{m=1, \dots, n}^{n=1, 2, \dots}$ – множина

$Z^{(N,N)}$ -чр-функцій та $Z^{(N,N)}$ -чр-предикатів. Подальші побудови показали, що справедлива

Теорема. $\sigma_{Z^{(N,N)}} = \left\{ =_Z, \nabla, \overset{+}{U}, \max, \{0\}, \uparrow, \downarrow, I_m^n \right\}_{m=1, \dots, n}^{n=1, 2, \dots}$ – породжуюча система

ППА $A_{Z^{(N,N)}}^{cp}$.

Крім зазначеного вище, досліджено семантичні суті синтаксичних конструкцій HDL-мов на прикладі мови Verilog і деякі інші їхні семантичні аспекти таких. У ході дослідження мови Verilog виділено такі синтаксичні конструкції: галудження *if*, галудження *case*, циклювання *repeat*, циклювання *while*, *for*, *forever*;

Їхні семантичні суті докладно описані та можуть розглядатись як композиції. Отже, викрито композиційну алгебру, яка лежить в основі мови Verilog, що в свою чергу дає змогу застосовувати механізм АПС для вирішення задач із розробки апаратного забезпечення. Слід зауважити, що не кожна з представлених синтаксичних конструкцій мови Verilog може бути втілена в апаратне забезпечення (бути синтезованою). Звісно, для розробки реального апаратного забезпечення необхідно використовувати конструкції, що синтезуються. Отже, множину синтаксичних конструкцій мови Verilog можна розбити на три групи: конструкції, що синтезуються; конструкції, що ігноруються при синтезі; конструкції, що не синтезуються.

Verilog та інші HDL-мови мають деякі обмеження щодо можливості синтезу деяких синтаксичних конструкцій. Для кожного синтезатора ці дані відрізняються, але їх більшість збігається. Цикли *forever* та *while* не синтезуються взагалі. Цикл *repeat* синтезується лиш у тому разі, коли вираз, який визначає кількість ітерацій циклу, може бути обчислений під час синтезу. Цикл *for* синтезується тільки в тому разі, якщо всі присвоєння для ітератору (індексу) циклу константні. Варто пам'ятати, що синтезована цифрова схема не має ні часу початку роботи, ні часу закінчення роботи – вона просто існує. Тому ігноруються операції, які розробник задає для виконання на початку її роботи (але вони не ігноруються під час симуляцій). Початкових значень регістрів теж не існує. Також ігноруються змінні, призначені для симуляції затримки виконання, котрі розробник може задавати для виразів.

Вирішення цих «незручностей» знаходиться поза межами Verilog. Для таких випадків, наприклад, є сигнал скидування, про подання якого та про реакцію на який має думати розробник. Саме сигнал скидування може бути умовним початком роботи схеми. При поданні тактового сигналу розробник задає рівномірний потік дискретного часу. Здебільшого цифрові електронні пристрої працюють саме в дискретному часі, який задається тактовим сигналом. Отже, необхідно розширити платформу проектування сигналами скидування та тактовим сигналом і промоделювати композиції, яких бракує, з використанням цих сигналів і синтаксичних структур, що синтезуються, що й зроблено в дисертації. Це дало змогу у наступному розділі імплементувати адаптивне процесональне середовище, яке дає змогу отримувати рішення в Verilog HDL.

За результатами проробленої роботи у розділі «**Імплементация адаптивного процесонального середовища**» розроблено адаптивне процесональне середовище для розробки апаратного та програмного забезпечень, яке дає змогу застосовувати та породжувати композиції й здійснювати декомпозицію задачі. Воно підтримує автоматичну генерацію коду задачі в мовах Сі та Verilog HDL і має графічний інтерфейс для декомпозиції задачі.

Під час розробки середовища розроблені графічна й текстова мови запису композицій. Вони дозволяють задавати та застосовувати функції, задавати та застосовувати композиції й застосовувати метакомпозиції. Записи в графічній мові відрізняються підвищеною сприйнятливістю.

Розглянемо БНФ текстової мови запису композицій. Ліва та права дужки, між якими містяться аргументи для аплікації функцій: $lFapply ::= <"(">$ та $rFapply ::= <")">$. Ліва та права дужки, між якими містяться аргументи для аплікації композицій: $lCapply ::= <"[">$ та $rCapply ::= <"]">$. Ліва та права дужки, між якими містяться аргументи для аплікації мета композицій: $lMapply ::= <"<">$ та $rMapply ::= <">">$.

Операція іменування $assign ::= <"=">$. Ціле число $integer ::= <цілі числа>$. Ім'я $name ::= <імена можуть складатися з цифр і літер, а починаються - завжди на літеру.>$

Аргументи функцій – $function_arguments ::= <integer>...$ Аргументи композицій – $composition_arguments = <<Capply> | <name>>...$ Аргументи метакомпозицій – $metacomp_arguments = <<Mapply> | <name>>...$

Застосування функцій $Fapply ::= <name><lFapply><function_arguments><rFapply>$, застосування композицій $Capply ::= <name><lCapply><composition_arguments><rCapply>$, застосування метакомпозицій $Mapply ::= <name><lMapply><metacomp_arguments><rMapply>$.

Іменування функцій, композицій, метакомпозицій: $valueAssignment ::= <name><assign><Fapply>$, $funcAssignment ::= <name><assign><Capply>$ $compAssignment ::= <name><assign><Mapply>$.

Щодо графічного синтаксису, то його суть коротко можна виразити рисунком:



Рис. 1. Елементи графічного синтаксису для задавання функцій та композицій.

На рис. 1 зображені блоки, які являють собою функції, композиції та метакомпозиції. Усі вони можуть бути об'єднані в ненаправлений ациклічний граф (дерево). Ребра графа представляють зв'язки між композиціями, метакомпозиціями та їхніми аргументами. У роді позначень можна

використовувати як ім'я відповідного елемента, так і схему його побудови, намальовану на місці для запису позначення.

Ефективність розробленого середовища показується з використанням емпіричної моделі оцінки затрат на розробку СОСОМО. Результати аналізу АПС показали, що затрати на розробку можуть бути зменшені на 68%.

Звісно, таке адаптивне процесональне середовище було б складно застосовувати без можливості інтеграції отриманих рішень в інформаційно-обчислювальні системи, що вже існують. Особливо це стосується апаратного забезпечення. Саме тому описано технологію розробки драйверів для апаратного забезпечення, отриманого за допомогою адаптивного процесонального середовища (адаптивного апаратного забезпечення, ААЗ), для операційної системи на базі ядра Linux. Є два способи отримати доступ до ААЗ з операційної системи: написати модуль ядра, який би міг працювати в режимі супервізора, зчитувати та записувати дані фізичної пам'яті й, водночас, представляти на спеціальній файловій системі `sysfs` пристрій; використати системні функції, які дають змогу відображати фізичну пам'ять у віртуальний простір адресу користувачької програми (`user-space`).

Ми зупинилися на першому варіанті як на найбільш широковживаному й відповідному нашій задачі. `sysfs` – спеціальна файлова система, яка створюється ядром Linux. Використовуючи віртуальні файли (атрибути), `sysfs` експортує інформацію про версії різних підсистем ядра, апаратні пристрої й пов'язані з ними драйвери в об'єкти файлової системи, які може читати та записувати користувач. До того ж, крім інформування користувачів і розробників ПЗ, `sysfs` використовується для конфігурування пристроїв і підсистем ядра. Користувачька модель ААЗ в `sysfs` містить $n+1$ файл, де кожен з n файлів відповідає за один вхід, а $n+1$ 'ий файл відповідає регістру пристрою з результатом. Вони нумеруються наступним чином `in1`, `in2`, `in3`, ..., `inn`, `res` (рис. 2). Кожен файл «`in`» доступний тільки для запису, а «`res`» - тільки для зчитування. Обчислення запускаються після того, як записаний останній вхідний регістр (`inn`). Усі ці файли розміщені в `/sys/kernel/mm/devicename` (рис. 2).

```

/sys
  kernel
    mm
      devicename
        in1
        in2
        ...
        inn
        res

```

Рис. 2. Представлення адаптивного апаратного забезпечення в `sysfs`.

Також розроблені засоби для автоматичної генерації модулів для САПР Quartus II Qsys, які значно спрощують процес розробки обчислювальної системи з використанням апаратного забезпечення, отриманого за допомогою адаптивних процесональних середовищ.

Висновки

У дисертаційній розроблений метод ефективної побудови композитосутнісних моделей якісного та надійного вирішення ІТ-задач, зокрема розробки апаратного та програмного забезпечень. Основним результатом дисертації є розроблений адаптивний підхід до побудови моделей ІТП. На його основі створена концепція єдиного концептуально цілісного процесонального середовища. Основу його складає біабстрактний розгляд ІТ-процесів, побудований на концепції відкрито-замкнених систем та композитосутнісного підходу, як прагматико-обумовленої конкретизації сутесутнісного підходу.

Головними результатами дисертації є:

1. Розроблена концептуально єдина *загальнопонятійна система* моделювання ІТП, на її основі створені змістовні засади АПС, які враховують активну роль суб'єкта в ІТП та підкреслюють важливість самого моделювання ІТП.
2. На основі загальнозмістовної понятійної системи АПС побудовані *логіко-математичні засади* АПС та композиційні основи АПС, що в значній мірі витікають із праць учнів композиційної школи програмування. Засади підкреслюють, що в основі ІТП є проєктивна функція, яка представляє рішення, а засоби побудови таких функцій являють собою алгебраїчні операції над ними (композиції).
3. На основі композиційної понятійної системи створені *композитосутнісні засади моделей АПС*. Показано, що АПС зводиться до ІТ-релятивізації універсального середовища інтеграції (УСІ). А саме середовище інтеграції є взаємодоповненням універсуму рішень, універсуму предметів, універсуму засобів побудови та засобів еволюційного розвитку.
4. Запропоновано прагматико обумовлену типізацію композицій. Досліджені фундаментивний та аплікативний класи композицій над абстрактними, поліарними та метаполіарними проєктивними функціями. Розроблений *метод композитосутнісної релятивізації* проєктивних функцій. Доведена *теорема про нерухому точку* проєктивних функцій.
5. *Досліджено семантичні характеристики* репрезентативних класів обчислюваних функцій над зліченими носіями, зокрема класом функцій та предикатів над записами. Також досліджені семантичні структури, що містяться в мові Verilog HDL.
6. Розроблено дослідну реалізацію АПС на мові програмування Python. Є можливість задавати композиції як текстовим синтаксисом, так і в діалоговому режимі через вибір відповідних редукцій розробником.

Базові набори функцій та композицій легко замінюються. Вирішено низку репрезентативних задач, які демонструють ефективну розробку з використанням АПС. Таке середовище, з огляду на своє походження, дає змогу остаточно вирішити питання верифікації рішень.

Одержані в дисертації нові результати знайшли застосування в навчальному процесі під час підготовки НТУУ “КПІ” фахівців у галузі радіоелектронного апаратобудування за спеціальністю “Виробництво електронних засобів” та можуть бути використані під час підготовки вищими навчальними закладами України фахівців із розробки ІТ-рішень, зокрема програмних та апаратних обчислювальних засобів, в тому числі, за спеціальностями “Основи побудови інформаційно-обчислювальних засобів інтеграції” та “Програмна інженерія”, а також використані в результатах НДР та впроваджені в організації ТОВ “Відео Інтернет Технології”, що підтверджується відповідними актами.

Список опублікованих праць за темою дисертації

- [1] T. L. Zakharchenko, "Adaptive hardware design," *Технологический аудит и резервы производства*, no. 5, pp. 23-31, 2015.
- [2] T. Zakharchenko, "Pragmatics dependent hardware design," in *Proceedings of the 2014 IEEE 34th International Scientific Conference "Electronics and nanotechnology"*, Kyiv, 2014.
- [3] Т. Л. Захарченко, «Исследование возможности реализации эффективной вычислительной архитектуры на реконфигурируемых логических устройствах,» *Вісник КНУТД*, № 1, pp. 35-39, 2014.
- [4] Т. Л. Захарченко, «Исследование возможности реализации эффективной вычислительной архитектуры на реконфигурируемых логических устройствах,» в *Збірник статей VI Міжнародної науково-технічної конференції молодих вчених «Електроніка-2013»*, Київ, 2013.
- [5] Т. Л. Захарченко, «Вирішення проблеми повноти в композиційному програмуванні,» в *Матеріали XIII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики»*, Київ, 2015.
- [6] Т. Л. Захарченко, «Швидкодіючий багатошаровий перцептрон, що гнучко масштабується». Україна Патент 72313, 10 08 2012.
- [7] Т. Л. Захарченко, І. В. Редько, Д. І. Редько та П. О. Яганов, «Примітивна програмна алгебра обчислюваних функцій над записами,» *Наукові вісті НТУУ «КПІ»*, № 2, pp. 29-40, 2015.
- [8] D. I. Redko, I. V. Redko, P. O. Yahanov and T. L. Zakharchenko, "Compositional basis in programmer activity," *Системні дослідження та інформаційні технології*, no. 4, pp. 83-96, 2015.
- [9] Т. Л. Захарченко та І. В. Редько, «Проблема повноти в класі функцій над записами, які зберігають денотати,» *Наукові вісті НТУУ «КПІ»*, № 5, pp. 23-30, 2015.
- [10] І. В. Редько, Д. І. Редько та Т. Л. Захарченко, *Концептологічні основи проектування*, Київ: Компринт, 2016.

АНОТАЦІЯ

Захарченко Т.Л. Композитосутнісні моделі адаптивних процесональних середовищ. – На правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 – інформаційні технології. – Інститут телекомунікацій і глобального інформаційного простору, Київ, 2018.

Дисертацію присвячено розв’язанню задачі ефективної побудови якісних та надійних моделей інформаційно-технологічних процесів, зокрема процесів розробки апаратного та програмного забезпечень. Запропоновано композитосутнісний підхід, який дає змогу моделювати такі процеси. Даний підхід є прагматико-обумовленою конкретизацією сутесутнісного підходу, який використовується в дескриптивних середовищах. Головну роль у композитосутнісному відношенні грає композиція, вона і є головним засобом побудови рішень у даному відході, що підтверджується доведеною прагматико-обумовленою конкретизацією теореми Тарського. Проведені ґрунтовні дослідження поняття композиції, а також розроблена дослідна реалізація адаптивного процесонального середовища, яка здатна породжувати як специфікації апаратних рішень, так і програмних.

Ключові слова: Адаптивне процесональне середовище, композиція, відкрито-замкнене середовище, сутесутнісне відношення, композитосутнісне відношення.

Abstract

Taras Zakharchenko. Composition-gist models of adaptive processional environments – qualification scientific paper as manuscript.

Candidate's of science (philosophy doctor) dissertation on specialty 05.13.06 "Information Technologies". - Institute of telecommunications and global information space, Kyiv, 2018.

The paper is about solving problem of effectiveness and quality of information technology activity, including design of software and hardware. It is proposed composition-gist approach, which allows to support modelling of IT-processes, including design of both software and hardware projects. Given approach is pragmatically-conditioned specialization of the substance-gist approach, which is used in descriptive environments. The main role in substance-gist relation is played by the composition, it is the main modelling tool in proposed approach, and this is verified by pragmatically-conditioned and proved specialization of the Tarsky theorem. Profound research of composition concept conducted and test implementation of adaptive processional environment developed. It is able to produce both software and hardware project specifications.

Keywords: Adaptive processional environment, composition, open-closed environment, subject-gist relation, composition-gist relation.