

Київський державний економічний університет

На правах рукопису

**АХЛАМОВ** Анатолій Геннадійович

**ОСНОВИ ТЕОРІЇ ТА МЕТОДИ АВТОМАТИЗОВАНОГО  
БУДУВАННЯ БАЗ ІНТЕЛЕКТУАЛЬНИХ ІМІТАЦІЙНИХ  
МОДЕЛЕЙ РЕГУЛЯРНОГО ВИКОРИСТАННЯ**

08.03.03. - "Інформаційні системи в і

Автореферат дисертації на здобуття  
ступеня доктора економічних

Київ 1995

0.4  
0:004



00376065 (R)

Дисертація є рукописом.  
державному економічному  
Інформаційних економічних

Науковий консультант - доктор економічних наук,  
професор СИТНИК В. Ф.

Офіційні опоненти: - д. т. н., проф. Ковлик Г. А.  
- д. е. н., в. н. с. Ревенко В. Л.  
- д. е. н., проф. Заброцький В. А.

Провідна організація - інститут економіки  
промисловості (м. Донецьк).

Захист дисертації відбудеться "9" листопада 1995 р. о  
17 годині на засіданні спеціалізованої вченої ради Д 01.53.02  
при Київському державному економічному університеті за адресою:  
м. Київ, проспект Перемоги, 54/1, ауд. 214.

З дисертацією можна ознайомитися у бібліотеці Київського  
державного економічного університету.

Автореферат розісланий "6" листопада 1995 р.

Вчений секретар  
спеціалізованої вченої ради

А. Д. Шаповал

### ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

**Актуальність теми.** Однією з характерних особливостей сучасного етапу науково-технічного прогресу є ускладнення структури виробничих систем та процесів їх керування. Ці обставини різко підвищують міру відповідальності за прийняття науково - обґрунтованих рішень, особливо в умовах постійного часового дефіциту і збільшення об'єму інформації, яка надходить для оперативного аналізу. Тому моделювання процесів прийняття рішень стає центральним напрямком автоматизації діяльності осіб, які приймають рішення.

Серед методів прикладного системного аналізу імітаційне моделювання на ЕОМ є одним з найбільш міцних інструментів дослідження, розробки складних систем і процесів, керування якими зв'язано з пошуком рішень в умовах невизначеності. В процесі експериментів з моделями на ЕОМ можна опрацьовувати та уточнювати засоби узгодження рішень, які приймаються у всіх аспектах: часовому, функціональному та організаційно-технологічному.

Для організації подібних експериментів потрібно створити програмні системи, які являють собою людино - машинні системи пошуку рішень з автоматизованим формуванням варіантів (альтернатив). Інформаційною основою подібних систем є банк імітаційних моделей регулярного вжитку об'єктів, які вивчаються і процесів, котрі постійно поповнюються новими, апроксимованими і вже вжитими раніш.

Ефективне застосування метода імітаційного моделювання в значній мірі залежить від оперативності розробки і втілення нових програм. Досвід показує, що розробка машинних мо-

делі є самим вузьким місцем у вирішенні комплексного завдання системного імітаційного моделювання і його ефективне вирішення можливе на засадах розробки нових засобів автоматизації процесів моделювання, які дозволяють перекласти як можна більше функцій конструювання моделей на ЕОМ.

На жаль, процеси формалізації і математичного описання об'єктів матеріального світу в теперішній час в недостатній мірі автоматизовані. Ця обставина призводить до того, що розробка кожної нової моделі потребує великої спільної роботи користувачів, математиків та програмістів.

На сьогоднішній день в значній мірі вичерпані ті резерви зростання продуктивності узгодженої праці фахівців різного профілю при конструюванні імітаційних моделей розрахованих на кінцевого користувача. Тому активно триває пошук нових принципів, методів і засобів вирішення подібних задач. Головна мета пошуку - забезпечити спеціаліста у даній конкретній галузі інструментом дослідження в праці, в якій він (користувач) зміг би прийняти активну участь на всіх етапах життєвого циклу баз інтелектуальних імітаційних моделей.

Під інтелектуальною імітаційною моделлю, в цій праці, будемо розуміти програмну імітаційну модель (втілену на мові моделювання), формування якої здійснюється з залученням методів та засобів штучного інтелекту. Крім того, ця модель повинна містити знання необхідні для автоматизації окремих етапів життєвого циклу моделі, як наприклад: модифікація, адаптація у рамках СППР, інтерпретація наслідків і т.п.

Тому в теперішній час актуальними є розробки принципово нових підходів до імітаційного моделювання - дружнього споживачеві, з одного боку, і досить міцного для розробки складних систем, з іншого боку. Необхідність розробки таких засобів уявляється ще більш очевидним в зв'язку з необхідністю

вживанням ПЕОМ в наукових дослідженнях, що надзвичайно поширило діапазон втілення засобів обчислювальної техніки і викликало необхідність регулярного і оперативного рішення прикладних задач (з використанням імітаційного моделювання) споживачами - непрофесіоналами.

**Мета та задачі праці.** Метою цієї праці є розробка основ теорії, методів проектування й споживання баз імітаційних моделей орієнтованих на експлуатацію програмних систем регулярного вжитку у складі АРМ та СППР.

У відповідності з поставленою метою задачі дисертаційної роботи сформульовані таким чином:

- на основі аналізу сучасних технологій імітаційного моделювання виявити найбільш перспективні напрями автоматизації, як окремих етапів побудови і використання імітаційних моделей, так і усього процесу в цілому;

- провести аналіз доступних систем спеціального математичного забезпечення, які підтримують як традиційні так і перспективні технології створення імітаційних моделей;

- розробити математичний формалізм описання моделей і на його підставі розробити методикку проектування складних моделей;

- на базі вищезазначеного формалізму розробити технологію проектування моделей, зокрема технологію модульного моделювання;

- в цілях перевірки основних концепцій розробити програмову інструментальну систему, яка підтримує запропоновану технологію проектування. Провести випробування інструментальної системи при розробці конкретних систем. Висновки досліджень узагальнити.

**Методи дослідження.** При аналізі і розробці технології композиційного імітаційного моделювання були широко застосовані методи теорії мереж Петрі і їх модифікацій, методи мультного та об'єктно - орієнтованого програмування, а також різні інші методи формалізації і уяви складних систем.

При розробці методології імітаційного моделювання був застосований синтетичний підхід, метою якого було поєднання позитивних якостей існуючих або проєктованих засобів застосування до даної предметної області..

Правильність застосування цих методів підтверджена практичною апробацією результатів дослідження.

**Наукова новизна.** В процесі праці над дисертацією був отриманий та використаний ряд результатів, які мають самостійне значення:

- підхід до оцінки систем імітаційного моделювання з точки зору формалізму надання знань про системи, що моделюються і технології розробки й експериментування з імітаційною моделлю;

- новий клас імітаційних моделей - інтелектуальні імітаційні моделі регулярного застосування;

- застосування мережового підходу до побудови систем імітаційного моделювання складних систем;

- формалізм модифікованих мереж Петрі як засіб представлення знань про змодельовані системи, які дозволяють описувати складну поведінку дискретної та комбінованої природи;

- концепція підмоделі, яка дозволяє створювати проблемно - орієнтоване оточення для користувачів - непрофесіоналів;

- технологічні заходи, які дозволяють ефективно використовувати концепцію підмоделі для створення систем імі-

таційного моделювання з просунутою проблемною орієнтацією;  
- метод оцінки складності програм імітаційних моделей.

**Практична цінність роботи.** Проведені в роботі дослідження покладені в основу реалізації засобів інтерактивного імітаційного моделювання. Розроблена методика дозволяє оперативно створювати моделі складних систем, залучаючи на більшості стадій моделювання обмежене число фахівців. Запропонована програмна інструментальна система ГІМН (Генератор Імітаційних Моделей), яка підтримує запропоновану технологію створення і використання імітаційних моделей. В цілому розроблений комплекс дозволяє значно скоротити час, який витрачається на створення імітаційних моделей призначених для регулярного використання у СППР у широкому колі предметних областей.

Впровадження результатів досліджень було здійснено в ряді організацій та підприємств, у тому числі у генеральній дирекції Укрсоцбанку (м. Київ), НВО "Темп", КВЕТУ і т.п.

На ряді об'єктів був роахований річний економічний ефект від впровадження результатів дисертаційної роботи.

Результати роботи увійшли у лекційні курси, які читались автором у КДЕУ, ОДЕУ та С/ПДУ в зв'язку з основною виробничою діяльністю у період з 1977 по 1995 роки.

**Апробація результатів дослідження.** В період з 1985 по 1994 роки основні результати досліджень, вищеданих у роботі регулярно доповідались та обговорювались на всесоюзних, міжнародних, республіканських конференціях, симпозиумах і семінарах, у тому числі на:

- Республіканських семінарах "Застосування математичних методів в економічних дослідженнях", 1981 р. "Інформатика і

надійність" (Одеса), 1987 - 1993 р., 'Перспективні технології програмування', 1990-1993 р., Наукової ради по проблемі "Кібернетика", ІК АН України, м. Київ;

- підсумкових наукових конференціях професорсько - викладацького складу Одеського інституту народного господарства (1986 - 1990 р.) та Київського економічного університету (1990 - 1993 р.);

- Всесоюзному науково - технічному семінарі "Якість програмних засобів", м. Калінін, 1992 р.;

- Всесоюзній школі - семінарі "Психологічна біоніка", м. Харків, 1990 р.;

- Всесоюзній школі-семінарі "Проектування автоматизованих систем контролю і керування складними об'єктами", м. Харків, 1990 р.;

- Всесоюзній школі - семінарі "Біоніка інтелекту", м. Харків, 1992 р.;

- Всесоюзній конференції "Інформаційні технології у моделюванні", Новосибірськ, 1991 р.;

- Всесоюзній конференції "Синтез, тестування, верифікація та налагодження програм", м. Рига, 1991 р.;

- Республіканській конференції "Інформаційна діяльність в нових умовах господарювання", м. Тбілісі, 1989 р.;

- Республіканський семінар "Створення та використання економічних інформаційних технологій при утралінні будівельним виробництвом", Львів, 1992 р.;

- Республіканській школі - семінарі "Сучасна програмна продукція", м. Одеса, 1989 р. та інші.

**Публікації.** Основні результати праці викладені більш ніж в п'ятидесяти надрукованих роботах загальним обсягом

понад 25 д. а., у тому числі у двох монографіях.

**Структура дисертації.** Праця складається зі вступу, семи глав, заключення, списку літератури з 239 найменувань і додатків. Основний текст викладений на 300 сторінках машинописного тексту і містять 55 малюнків і 14 таблиць.

## З М І С Т   Р О Б О Т И

У вступі обґрунтовується актуальність теми, визначаються цілі роботи, наукова новизна і практична цінність отриманих в дисертації результатів.

**Перша глава** дисертації присвячена короткому огляду сучасних засобів імітаційного моделювання. В цій главі ставляться задачі визначити специфіку сучасного стану імітаційного моделювання, основні проблеми імітаційного моделювання на сучасному етапі і ключові засоби вирішення цих проблем.

Аналіз еволюції застосування методу імітаційного моделювання засвідчив про те, що вимоги до цього методу стали більш жорсткими:

- реальні системи, які моделюються, мають в собі компоненти різної природи. Це породжує необхідність розробки універсальних парадигм моделювання;

- підвишена відповідальність до результатів моделювання. Це породжує потребу підвищення адекватності моделі системи;

- необхідність в оперативних змінах моделі як відповідь на зміни в системі;

- масовість та регулярність застосування методу імітаційного моделювання - необхідність розробки засобів, доступних користувачеві - фахівцю.

Аналіз характеру використання обчислювальної техніки в різних предметних галузях та особливо в імітаційному моделюванні дозволяє виділити три рівня наданих кінцевому користувачу послуг:

- результати дослідження (розрахунків);
- моделі систем, які придатні до експериментування з ними;
- проблемноорієнтовані системи моделювання, які дозволяють фахівцю предметної області самостійно проводити увесь цикл імітаційного дослідження.

Чим вище рівень системи імітаційного моделювання, тим різноманітними повинні бути вимоги до знань, необхідних для створення і застосування такої системи. В роботі висуваються чотири наступних рівня компетентності:

- рівень фахівця предметної галузі, спроможного більш менш формалізовано викласти закономірності функціонування системи, яка досліджується та цілі імітаційного дослідження;
- рівень модельєра - фахівця, спроможного зпроектувати специфікацію системи, яка дана фахівцем предметної галузі в концептуальний базис деякої системи моделювання. Передає трансформовану специфікацію дослідженої предметної галузі фахівцю з розробки систем імітаційного моделювання;
- рівень фахівця з розробки систем імітаційного моделювання, що трансформує специфікацію предметної галузі в специфікацію, що визначає "індивідуалізовану проблемно - орієнтовану систему імітаційного моделювання";
- рівень програміста - реалізує вищевиділену специфікацію в систему імітаційного моделювання, користуючись метасистемою генерації систем імітаційного моделювання.

Ці чотири рівня компетентності можуть розподілятися в системі між створювачами системи та її користувачами різними засобами.

Еволюція засобів імітаційного моделювання йшла шляхом поетапного представлення зазначених сервісних можливостей:

- використання для потреб імітаційного моделювання засобів традиційних мов програмування;

- створення спеціалізованих мов моделювання та систем імітаційного моделювання на їх основі;

- створення систем імітаційного моделювання, в яких процес розробки моделі не зв'язаний з її текстуальним представленням на якій - небудь мові імітаційного моделювання. В цих системах увесь життєвий цикл моделі звичайно опирається на вбудований формалізм представлення знань про системи та зв'язаний з ним підхід до моделювання (дискретної, безперервної, комбінованої та ін.) складних економічних систем. Для користувача моделі в цих системах будуються безпосередньо зі специфікацій системи.

Відмічені складності використання традиційних мов моделювання: складність освоєння мови, складність валідації моделей та тестування програм імітаційного моделювання, відсутність міцних технологій, наприклад, підтримки модульного моделювання.

Більш докладно розглянемо системи імітаційного моделювання третього типу. Проаналізовані причини появи цього типу систем імітаційного моделювання. Важливий висновок заключається в тому, що засоби імітаційного моделювання розвиваються в напрямку підвищення їх "узагальноної ізоморфності", тобто властивості зберігати в моделі найбільше число рис системи - оригіналу. Чим сильніше "ізоморфність" представлення системи і моделі, тим кращі якісні характеристики моделі. Така ізоморфність повинна бути забезпечена узагальне-

ним в системі імітаційного моделювання формалізмом представлення знань про системи.

З цієї точки зору розглядається ряд систем третього виду ("немовні системи").

Одним з популярних засобів представлення знань про системи є мережі Петрі та їх багаточисельні версії. Безпосереднім здобуттям мереж Петрі для імітаційного моделювання є:

- мінімальність і ортогональність концептуального базису (поняття "умова", "постумова");

- висока структурна репрезентативність графічного представлення МП - збереження в графічній формі МП структури оригінальної системи, що робить МП "прозорим" для користувача засобом представлення.

Однак в якості самостійного та повного формалізму представлення знань про системи мереж Петрі практично не використовується.

При розгляді існуючих модифікацій МП робиться висновок про їх недостатню міцність для потреб представлення складних систем з різноманітною логікою взаємодій компонентів. Причому мається на увазі використання МП безпосередньо в якості мови для описання усіх закономірностей функціонування системи. Ясно, що виступаючи в такій ролі, МП втрачають якості, які дозволяють аналітично досліджувати системи, які представлені з їх допомогою.

Другим обширним класом систем "немовного" імітаційного моделювання є системи, які засновані на засобах обробки знань, які стосуються теорії штучного інтелекту. Крім загальноприйнятних формалізмів (продукційні правила, фрейми, семантичні мережі, логічні висновки тощо), в імітаційному моделюванні використовуються гібридні формалізми, наприклад, "логічне програмування + дискретне моделювання". В якості

показового представлення такого формалізму розглядається формалізм прийнятий у системі T/TS - PROLOG.

В результаті виконаного в першій главі аналізу робиться висновок про те, що ключевим моментом у вирішенні багатьох проблем розвитку систем імітаційного моделювання є розробка формалізму представлення знань про систему:

- зберігаючого "узагальнену ізоморфність" моделі та системи;

- допускою застосування міцних технологічних прийомів проектування (в першу чергу модульного).

**Друга глава** являє собою аналіз програмного забезпечення імітаційного моделювання на ПЕОМ, розглядаються ті зміни, які відбулись в ПЗ імітаційного моделювання, технології його розробки і в вимогах до Програм Імітаційного Моделювання (ПІМ) в зв'язку з переходом на новий клас обчислювальної техніки

Аналіз публікацій вітчизняних та зарубіжних авторів показав, що до ПЗ ІМ на сучасному етапі подаються високі вимоги, такі як:

- автоматична підтримка усіх етапів процесу створення моделі; забезпечення модульного підходу при проектуванні ІМ;

- об'єднання процесу моделювання з можливостями створення звітів, графічного відображення та аналізу результатів;

- мінімізації потреб в технічних знаннях в галузі використання комп'ютерів;

- можливість роботи як аналітикам - розробникам моделей, також і користувачам - фахівцям в конкретній предметній галузі;

- наявність засобів інтерпретації результатів моделювання до потреб осіб, які приймають рішення.

Відмічається, що якість ПІМ забезпечується як коректністю та повнотою концептуальної моделі, так і технологією розробки ПІМ, а також якістю і надійністю тих засобів, з допомогою котрих створюються ПІМ. Не дивлячись на підвищену кількість засобів до розробки ПІМ, їх різноманітність та переваги (простота використання, збільшення швидкості розробки ПІМ, зручне меню, підказки у реальному масштабі часу, багатокольорова графіка, анімація та ін.) існує ряд недоліків, які не дозволяють у цей час гарантувати надійність, якість, необхідний рівень деталізації та високу швидкість виконання великих та складних ПІМ.

Одним з факторів, які впливають на якість ПІМ, є трудові затрати. Від того, наскільки пророблена концептуальна модель, яка трудоемкість розробки, наладки і налагодження залежить якість ПІМ в цілому. Трудомісткість створення в найбільшому ступеню визначається складністю моделі. Програмна реалізація моделі є найбільш доступним компонентом до кількісного аналізу складності. Оскільки у цей час не існує методів зміни складності даного класу ПЗ, формуються цілі та задачі дослідження.

**Третя глава** присвячена розробці формалізму для представлення знань про системи, на базі яких можуть бути втілені ефективні інформаційні технології імітаційного моделювання. Сформульовані потреби до такого формалізму:

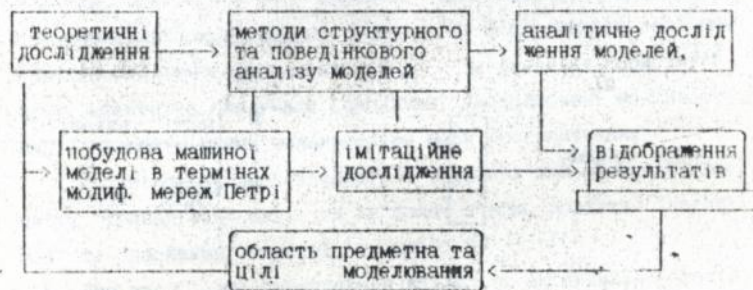
- дати користувачеві можливість вичерпно описувати змодельовану систему безпосередньо термінами цього формалізму: це означає, що терміни цього формалізму повинні безпосередньо відповідати термінам предметної області;

- підтримувати деяку регулярну технологію використання, яка дозволила б інтенсифікувати процес побудови (зокрема,

можливість модульного моделювання, автоматизації перевірки адекватності моделі і т. п.);

- забезпечити дружній інтерфейс споживача - фахівця предметної області з системи імітаційного моделювання, базуючись на цьому формалізмові, наприклад, шляхом візуалізації її процесів на різних стадіях життєвого циклу імітаційної моделі.

Застосування в системі імітаційного моделювання такого формалізму описання знань дозволяє зменшити кількість процесів, виконаних в технологічній схемі системного моделювання (див. мал. 1), що підвищує його ефективність та покращує якість характеристик здобутого програмного продукту. Так, наприклад, зменшується ймовірність втрати адекватності між моделю та модельованою системою.



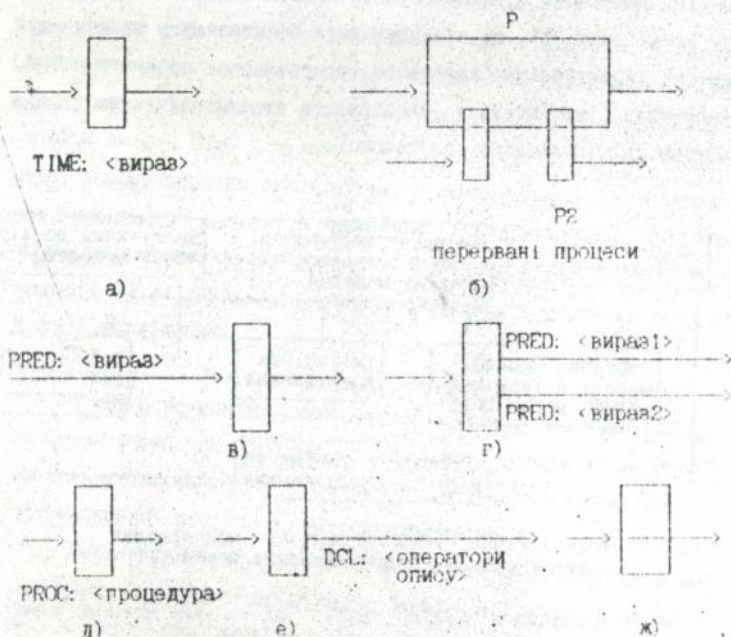
Мал. 1. Технологічна схема системного моделювання з використанням модифікованих мереж Петрі.

Як показано в першій главі, ці та інші корисні властивості формалізму представлення знань про систему залежать від його здібності зберігати "ізоморфність" моделі та системи. Для того, щоб розробити саме такий формалізм, треба вивести з достатньо універсального визначення поняття "система" та "модель".

Далі, в якості такого формалізму запроваджуються модифіковані мереж Петрі (ММП), які в поширенні традиційних мереж Петрі по наступних напрямках:

- індивідуалізація фішек;
- предикатні входні та вихідні дуги;
- часові переходи;
- обривні переходи;
- процедурні переходи.

Кожне з цих розширень має своє графічне зображення (див. мал. 2).



Мал. 2. Графічне зображення елементів ММП:  
 а) часовий перехід; б) P - перериваний перехід;  
 P1 - висволяючий перехід; P2 - звільнюючий перехід;  
 в) перехід з входнім предикатом; г) перехід з двома вихідними предикатами; д) виконуваний процедурний перехід; е) описуваний процедурний перехід; ж) прямі зображення мереж перехід фішки з атрибутами.

Індивідуалізація фішек являє собою найбільш суттєвий крок до того, що робить ММП природнім формалізмом уявлення систем. Індивідуалізація заключається у тому, що фішки мереж перестають бути просто індикаторами виконання умов, які наводять місця. Індивідуалізовані фішки набувають набір атрибутів, значення яких змінюється з часом внаслідок виконання переходів. Атрибути І-фішек беруть участь у:

- виразах, які задають тривалість виконання переходів;
- предікатах вхідних та вихідних дуг;
- процедурах переходів.

Динаміка ММП тепер подає у вигляді потоків І-фішек. Для збереження цілісності таких потоків у ММП реалізовані різні варіанти перенесення І-фішек через перехід. Одною з рис ММП, яка робить їх доступним і міцним засобом описування систем є їх заглиблення у середовище зрощеної версії мови PASCAL. Це означає, що багато які елементи ММП (часові затримання, предікати дуг, процедури переходів) формулюються засобами звичайного підмноження мови PASCAL. Це дозволяє користувачеві ефективно сполучати споруджені змодельовані можливості ММП з алгоритмічними можливостями мови програмування.

Часові переходи дозволяють в термінах ММП описувати поняття "тривалість" події та зв'язані з ним поняття "активний" та "пасивний" перехід (процес).

Для того, щоб забезпечити взаємодію активних переходів (наприклад, пріоритетне обслуговування), необхідно ввести в можливості ММП механізм переривів, для чого знадобиться ще два типи переходів: пориваючі та збільнюючі. Перехід, відносно якого виконуються дії з перерви та виволення, називається перерваним.

Процедурні переходи дозволяють модифікувати атрибути І-фішек та загальносистемні атрибути. При запровадженні процедур переходів виникає проблема: скільки разів виконується процедура переходу у зв'язі з атрибутами яких І-фішек вона

оперує. Для однозначного вирішення цієї проблеми на ММП покладено наступне обмеження - перехід може мати процедуру або коли у нього тільки одне вхідне місце та воно утримує І-фішки, або коли його вхідні місця не утримують І-фішки. Більш складні варіанти переходів легко "канонізуються" до цих двох.

Предикатні вхідні дуги визначають умови виконання переходів, більш складні, ніж просто наявність достатнього числа фішек у вхідних місцях.

Предикатні вхідні дуги дозволяють обрати шлях, по якому будуть спрямовані фішки після виконання переходу. Для того, щоб предикати вихідних дуг не "закрили" фішки в переході, запроваджується вимога логічної повноти предикатів вихідних дуг. Тобто  $p_1 \vee p_2 \vee \dots \vee p_n = \text{true}$ , де  $p_i$  - предикати вихідних дуг.

Далі запроваджується формальне значення ММП. ММП  $S$  являє собою кортеж:

$(P, T, F, M, W, FL, DCL, P\_IN_n, P\_IN_i, P\_OUT_n, P\_OUT_i, ATTR)$ ,

де

- P - безліч місць ММП;
- T - безліч переходів ММП;
- F - відношення інцидентності:  $F \subseteq P \times T \times T \times P \times M$  - функція розмітки ММП:  $M : P \rightarrow \{f\}$ , где  $\{f\}$  - безліч фішок;
- W - стислість дуг:  $W : F \rightarrow N \cup \{0\}$ ;
- DCL - функція опису І-фішок;
- ATTR - безліч загальносистемних атрибутів (у тому числі системний час - time).
- $P\_IN_n$  - вхідні предикати до місць, які не утримують І-фішки;
- $P\_IN_i$  - вхідні предикати до місць, які утримують І-фішки;
- $P\_OUT_n$  - вхідні предикати до місць, які не утримують І-фішки;
- $P\_OUT_i$  - вхідні предикати до місць, які утримують І-фішки;
- FL - відношення потоку до І-фішек, задає дерево місць, в які можуть бути спрямовані І-фішки у відповідності з предикатами з  $P\_OUT$ .

Перелічені компоненти визначають статистику ММП - тобто стан ММП в різних моментах часу. Крім того визначається ди-

СНП

НА

наміка ММП - тобто правило змін стану ММП при спрацюванні переходів.

Для цього безліч переходів  $T$  поділяється на безліч: пасивні та активні; перервані та вільні.

Для переходів ММП визначаються умови: активізації, пасивізації, спрацювання і на їх основі формулюється правило спрацювання переходу.

Четверта глава присвячена розробці на базі ММП концепції модульного моделювання. Досліджується питання про специфіку поняття "підмодель" в імітаційному моделюванні, в порівнянні з поняттям "підпрограма" в програмуванні. Аналізуються причини труднощів введення підмоделей в імітаційному моделюванні. Робиться висновок про те, що головною перешкодою в цьому напрямку є відсутність подібного формалізму для опису знань про системи. Аналіз поняття складні системи приводять до висновку, що найбільшу користь введення модульного моделювання може принести для задач моделювання "великих" систем. До "складних" систем модульне моделювання підходить в меншій мірі.

Наводяться рівні класифікації поняття "підмодель".

1. "Технологічна" класифікація підмоделей: абстрактна, типова, конкретна.

2. Класифікація, заснована на синхронізації входу та виходу підмоделі: підпрограма, синхронізована до входу та виходу, синхронізована до виходу, синхронізована до виходу та асинхронна.

Далі показано, як остання абстрактна класифікація може бути конкретизована на підставі ММП.

Внаслідок отримуємо визначення ММП - підмоделі (ММП-ПМ) у вигляді п'ятірки:

$S = \langle I, M, O, ACT, PARM \rangle$ , де

$I = \{ p \mid p \in P, x \in T: x \in F p \}$  - безліч вхідних місць

ММП ПМ;

$M = \{ p \mid p \in P, x \in T: x \in F p, y \in T: p \in F y \}$  - безліч

внутрішніх видимих місць ММП ПМ;

$O = \{ p \mid p \in P, y \in T: p \in F y \}$  - безліч вихідних місць

ММП ПМ;

$ACT = \{ act \}$  - безліч видимих процесів ММП ПМ;

$act = \langle name_{act}, t_{ast}, I_{ast}, O_{ast} \rangle$ ;

$name_{ast}$  - ім'я уявного процесу;

$t_{ast}$  - перехід, який відповідає видимого процесу;

$I_{ast}$  - безліч вхідних місць видимого процесу;

$O_{ast}$  - безліч вхідних місць видимого процесу.

$PARM$  - безліч параметрів ММП ПМ.

Крім відношення ММП ПМ до ефективного введення модульного моделювання вимагається визначити ряд корисних операцій комплексції ММП ПМ. В главі на підставі прикладів функціонування цеху неформально описуються, а потім і формально визначаються наступні операції комплексції підмоделей: стикування, розширки, злиття, вставки і накладання. Перші три операції підтримують технологію розробки моделей методом "знизу-вгору". Для реалізації технології "з гори - вниз" може використовуватись операція "вставка". Операція "накладання" враховує специфічні ситуації, які виникають при розробці моделей.

Усі операції комплексції реалізувались на базі операції ототожнення місць ММП. Наводиться формальне відзначення цієї операції.

Крім поняття "підмодель" і операцій комплексції підмоделей, необхідно забезпечити зручний інтерфейс користувача з системою імітаційного моделювання. Для цього розроблена мова опису підмоделей, яка описана у вигляді правил БНФ. Підмодель на цій мові зображується послідовністю пропозицій, аналогічних продукційним правилам і має два рівня опису: зовнішній - для користувача; внутрішній - для інтерпретатора ММП. Розроблена процедура трансляції зовнішнього представлення у внутрішнє. Це дозволяє вести розробку підмоделі в цілому в термінах речень мови опису підмоделей.

Реалізований також режим побудови підмоделей з допомогою меню - орієнтованого діалогу.

Таким чином, запропонований формалізм ММП є базою для введення міцних технологічних прийомів, в першу чергу модульного моделювання.

**В п'ятій главі** ставиться і обґрунтовується важливе у теоретичному й практичному відношенні питання побудови генератора GPSS - програм, визначених для автоматизації процесу побудови імітаційних моделей.

Відною інформацією до генерації програмних моделей є метауявлення задачі з неявним виразом алгоритму. Така інформація може бути отримана, наприклад, внаслідок роботи користувача з системою подання знань засобами ММП, які відображають дослідну предметну галузь в списочні структури, розділені на три рівні. До першого рівня відноситься головний список, який виконує роль диспетчера, до другого - набір списків (структур), які об'єднують адресні дані про атрибути

в списках третього рівня, складених з вказаних списків: стаціонарних об'єктів, розподілень та дій.

В якості математичного апарату генератора GPSS програм доцільно використовувати добре розроблений теоретичний клас народжених граматики Хомського.

Хай породжуюча граMATика задана четвіркою:

$G = (U, V_N, W, P)$ , де до даної задачі:

$U = \{U_i \mid i=1, \dots, m\}$  - кінцева безліч термінальних

символів, зображених сукупністю літер латинського алфавіту, набір цифр, спеціальних символів зв'язку, з яких будуються об'єкти GPSS-програм.

$V_N = \{V_i \mid i=1, \dots, n\}$  - кінцева безліч нетермі-

нальних символів, якими позначаються класи вхідних елементів або ланцюжків початкових елементів, а також, в окремих випадках, деякі спеціальні елементи;

$\bar{V}_N$  - початковий символ, який позначає

клас тих мовних об'єктів, для яких призначається дана граMATика. В даному випадку в якості  $W$  обирається "GPSSPROG";

$P = \{x_i \rightarrow h_i \mid i=1, 2, \dots, l\}$  - кінцева безліч правил,

де  $x_i, h_i$  - ланцюжки,

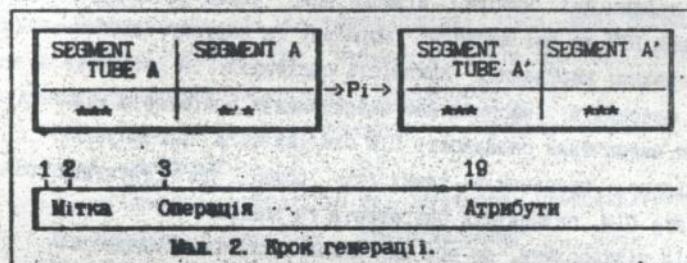
вибудовані в термінальних і нетермінальних символів. Кожне правило у формальному вигляді може бути описане наступним чином:

<ПРАВИЛО ГЕНЕРАЦІЇ> := <ЛІВА ЧАСТИНА> → <ПРАВА ЧАСТИНА>

<ЛІВА ЧАСТИНА> := <НЕТЕРМІНАЛЬНИЙ СИМВОЛ> |  
<НЕТЕРМІНАЛЬНИЙ СИМВОЛ> (<СПЕЦИФІКАЦІЯ УМОВ>)

<ПРАВА ЧАСТИНА> := <СЕРМЕНТ> | <ПРАВА ЧАСТИНА> <СЕРМЕНТ> і т.п.

В загальному випадку ліва частина правила визначає ситуацію, яка повинна існувати для того, щоб можна було застосувати це правило, а права частина інформує про те, що необхідно виконати при виводженні GPSS - програми. Процес, який перевіряє умови існування правила й виконує відповідну дію, будемо називати кроком генерації. На мал. 2. подане графічне зображення кроку генерації.



Мал. 2. Крок генерації.

У лівому верхньому куту схематично відображено увесь об'єм даних перед застосуванням правила P<sub>i</sub>, а у правому - об'єм перетвореної інформації. Послідовне застосування правил забезпечує видання повних ланцюжків, складених тільки з термінальних символів. Вивід ланцюжка триває до появи підсумкової продукції, яка відповідає конкретній GPSS програмі. В нижній частині малюнка визначено поле, в яке заноситься частина відповідного ланцюжка, яка відповідає повному блоку (оператора) мови GPSS. Такий спосіб представлення виві-

ду дозволяє відобразити скорочений запис декількох правил.

Ідея народження програмної моделі складається в виро-  
щення дерева граматичного розбору, який відбудовує синтак-  
сичну структуру фрази. На відміну від традиційного підходу,  
роблячи класифікацію фрази семантики по рівнях, можна уникну-  
ти деяких форм синтаксичної невизначеності.

Весь процес генерації GPSS програми представлений на  
прикладі.

В шостій главі розглядаються питання вимірювання склад-  
ності програм імітаційних моделей (ПІМ).

Аналізуючи особливості ЯІМ та ПІМ обґрунтовується дум-  
ка, чому неможна застосовувати традиційні метрики відносно  
цього програмного забезпечення.

До особливостей ПІМ, які не дозволяють механічно пере-  
носити традиційні методи, відносяться: чітке структурне  
розділення ПІМ на дві частини: модельну та керуючу, особли-  
вості передачі керування, наявність системних та користу-  
вальних операндів, інформаційне навантаження операторів ЯІМ.

Для визначення складності ПІМ аналізується пристосуван-  
ня декількох показників таких як: розмір та структурна  
складність ПІМ, складність операторів та даних.

Слід підкреслити, що при вимірюванні ступеню складності  
використовується абсолютна шкала оцінок, яка дозволяє кла-  
сифікувати об'єкти та отримати значення показників у реально  
вимірюваних фізичних величинах, а також відносна шкала, а  
допомогою якої можна вирааити, наскільки далеко знаходяться  
об'єкти від невимірної характеристики (еталону). В якості  
еталону оцінки складності ПІМ використовуються показники для  
реалізації простої системи масового обслуговування з чергою  
та з одним пристроєм на даному ЯІМ.

Розмір ПІМ включає в себе деякі з відомих метрик Холс-  
теда: довжина та об'єм, причому найбільш цікавий для аналізу

ПІМ подають відносні розміри цих показників. Головна проблема постає в розробці методології визначення головних показників (кількість унікальних операторів та операндів, загальна кількість операторів та операндів) до ПІМ на ЯІМ. При аналізі операторів бвиявляють ті елементи ЯІМ, які впливають або на перетворення даних, або на порядок використання даних, або на процес передавання керування.

Методи визначення складності розглядаються на прикладі ПІМ, які використані на GPSS, тому, що GPSS - є однією з найбільш поширених мов моделювання у тому числі і на ПЕОМ.

Для визначення структурної складності використовується диференційне циломатичне число, яке може бути визначене на базі графу потоку ПІМ  $H = (N, L)$ , який складається з двох кількостей:  $N$  - безліч вузлів та  $L$  - безліч ребер. Кожному вузлу графу  $H$  програми  $P$  відповідає або описання  $d_i$  або, оператор  $p_i$ . Обов'язкових описань в ЯІМ немає, тому вважається, що перемінна або константа описані, коли вони перший раз з'явилися у тексті ПІМ. Кількість вузлів дорівнює сумі кількості констант, перемінних і операторів.

Ребро  $L_i = (n_i, n_j)$  присутнє у графі потоку  $H$  в наступних випадках:

- при передачі управління від оператора, представленого  $n_i$ , безпосередньо оператору, представленому  $n_j$  ( $n_i$ ;  $n_j$  - вузли операторів);

- при виконанні оператора, представленого  $n_i$ , якщо можлива зміна значення перемінної, яка представлена  $n_j$ , або використання значення перемінної, представлені  $n_j$  ( $n_i$  - вузлом оператора,  $n_j$  - вузол описання).

Таким чином, ребро  $(n_i, n_j)$  між вузлами операторів указує на те, що вузол  $n_i$  безпосередньо "впливає" на вузол  $n_j$ .

Кількість вхідних вузлів  $N_p$  складається з двох компонент:  $N_c$  - кількість констант в позиціях стандартних числових атрибутів, які використовуються у програмі  $P$ ;

$N_o$  - кількість вхідних операторів, які забезпечують

введення транзактив у модель. Вихідними вузлами є ті оператори, які виводять транзакти з моделі. Число вихідних вузлів позначимо як  $N_w$ .

Диференціальна цикломатична складність ПІМ,  $F(P)$ :

$$F(P) = L - N + N_p + N_w, \quad (1) \text{ де}$$

$L$  - кількість ребер у графі  $H$ ;

$N$  - кількість вузлів у графі  $H$ ;

$N_p$  - кількість вхідних вузлів у графі  $H$ ;

$N_w$  - кількість вихідних вузлів у графі  $H$ .

Порівняння (1) може бути перетворено таким чином, щоб диференціальна цикломатична складність розраховувалась тільки з тексту ПІМ, без потреби створювати граф.

Запровадимо наступні позначення:

$A(p)$  - кількість гілок, які починаються у вузлі оператора  $p$ ;

$D(d)$  - кількість операторів, які змінюють значення імені, представленого описанням  $d$ , або які використовують це значення.

$$F(P) = \sum_{i=1}^n (A(p_i) - 1) + \sum_{j=1}^m (D(d_j) - 1) + N_c + N_t + N_w$$

Диференціальне цикломатичне число відображає складність передачі керування між блоками ПІМ та передачі даних між блоками (тобто відображають "вплив" вузлів графа потоку один на одного). Основний вклад в розмір диференціального числа вносять: кількість розгалужень у ПІМ, кількість вхідних та вихідних вузлів.

Цей показник структурної складності показує наскільки структурна складність ПІМ, що аналізується перевищує структурну складність еталонної ПІМ.

Крім того, у роботі позначені методи, які забезпечують

розрахунок складності операторів, даних ПІМ і в кінці отримання узагальненого показника складності.

В сьомій главі розглядаються питання практичного застосування розроблених методів. Описані основні принципи реалізації програмно-інструментальної системи імітаційного моделювання "ГІМН".

Засобами системи "ГІМН" був реалізований базис підмоделей, який обіймає основні моделюючі можливості мови GPSS. Це дозволило суттєво прискорити розробку моделей, які добре передають в термінах транзактноорієнтованого підходу. При цьому результати проведених розробок можуть зберігатися у вигляді підмоделей високого рівня, що значно підсилює моделюючу міцність системи.

Базис підмоделей. У початковий базис підмоделей увійдуть підмоделі, які реалізують засоби GPSS, зв'язані з блоками: GENERATE, TERMINATE, ADVANCE, ASSIGN, TRANSFER, TEST, GATE, SPLIT, ASSEMBLE, GATHER, MATCH, HELP, SAVEVALUE, 'MSAVEVALUE, LOGIC, SEIZE, RELEASE, PREEMPT, RETURN, ENTER, LEAVE. Природно, що при цьому слід мати можливість звертати до "СЧА". Як видно з таблиці 2 до реалізації необхідних СЧА достатньо використовувати дві втілені функції: CONT(N) - кількість фішок в місці N і CUR\_TIME - поточне значення відносно часу. Усі останні СЧА реалізуються (моделюються) на підставі цих двох спеціальних функцій з допомогою засобів Модифікованих мереж Петрі.

На підставі розробленого базису підмоделей були вирішені задачі моделювання об'єктів з метою використання моделей в інструментальних системах, які орієнтовані на кінцевого користувача. Так, наприклад, в дисертації докладно розглядаються приклади користування баз моделей у СППР Укрсоцбанку та АРМ начальника складального цеху машинобудівного заводу.

Таб. 2.

ОБ'ЄКТ	ПАРАМЕТР		МЕТОД
Пристрій	F	стан зайнятості пристрою	вбудовується
Пристрій	FR	коефіцієнт використання	моделюється
Пристрій	FC	загальне число входів у пристрої	моделюється
Пристрій	FT	середній час користування	моделюється
Функції	FN	обчислене значення	моделюється
Групи		відсутні	
Матриці чарун.	$M \begin{Bmatrix} H \\ X \\ B \end{Bmatrix}$	(a, b)	моделюється
Чергі		відсутні	
Чарунки	$X \begin{Bmatrix} H \\ B \end{Bmatrix}$	поточний зміст	моделюється
Пам'яті	S	поточний зміст пам'яті	моделюється
	R	число вільних одиниць	вбудовується
	SR	коефіцієнт використання	моделюється
	SA	середнє значення	моделюється
	SM	максимальне значення	моделюється
	SC	число входів	моделюється
	ST	середній час використання	моделюється
Системні атрибути	RN	випадкове число	моделюється
	C	поточний зміст відносного часу	вбудовується

У заключенні коротко сформулюємо основні результати дослідження.

Теоретичне узагальнення результатів конкретних робіт щодо створення спеціального класу баз програмних імітаційних моделей регулярного використання, які засновані на застосуванні сучасних інформаційних технологій, у тому числі з використанням засобів штучного інтелекту, дозволили отримати такі результати:

1. Визначені основні риси програмного забезпечення імітаційного моделювання на ПЕОМ, призначеного для регулярного використання як специфічного виду товарної продукції, запропонований новий підхід, який забезпечує автоматизацію процесу створення моделей, визначені шляхи підвищення продуктивності праці розробників моделей.

2. При розробці методології створення систем комплексної автоматизації проектування імітаційних моделей в якості ба-

вової прийнята концепція, яка визначає проектування моделі деякого складного об'єкту як послідовний перехід від первинного описання об'єкту через проміжні моделі до кінцевої моделі. Визначена структура сукупності моделей, які формуються та використовуються у процесі автоматизованого отримання кінцевого продукту (моделі) з заданими властивостями.

3. Обґрунтований вибір та розроблені мовні засоби щодо описання метамоделі. В якості основного засобу запропоновано використовувати апарат модифікованих мереж Петрі, які задовольняють основні вимоги, що пред'являються до засобів специфікації моделей (однозначність, точність, адекватність, модифікованість, простота засвоєння, наглядність та інше).

4. Розроблені методи та алгоритми, які використовуються при побудові переробників інформації необхідної щодо створення моделей: первинного описання концептуальної моделі, потім у логічну модель, далі у програмно-алгоритмічну, і нарешті, в програмну імітаційну модель на заданій мові моделювання.

5. Розроблена сукупність методів та алгоритмів автоматизованого конструювання програм імітаційних моделей, тобто побудови логічних схем створених програмних продуктів, які утворюються, описання яких є важливою частиною вхідної інформації щодо генератора програм, який формує імітаційну модель з заданими якостями.

6. Розроблені принципи, методи та алгоритми побудови системи атестації імітаційних моделей реалізованих на заданій мові моделювання. Запропонована система забезпечує контроль програмної моделі починаючи від перевірки коректності специфікацій до тестування створених програм.

7. Розроблена архітектура та схема функціонування системи машинного проектування програмних імітаційних моделей у інтерактивному режимі. Розроблена структура основних функці-

ональних підсистем та їх конструктивні елементи.

8. Розроблена архітектура генератора програмних імітаційних моделей на мовах моделювання, забезпечення його виконання у системному та автономному режимах. Запропоновані дослідження генерації на двох рівнях вхідних мов генератора програм моделей: простих елементів та інтелектуальних підмоделей.

9. Досліджувані методи автоматизованого формування імітаційних моделей на підставі інтелектуальних підмоделей, які накопичуються в інформаційній базі системи автоматизованого проектування. Розроблена структура, схема функціонування та алгоритми основних блоків системи ГІМН.

10. Узагальнений досвід використання запропонованих у роботі методів та засобів автоматизації процесу побудови програмних імітаційних моделей при створенні ряду СППР, які забезпечують: управління трудовими процесами на машинобудівному підприємстві, управління великим комерційним банком та його філіями і т.п.

Основні результати дисертації були опубліковані в наступних виданнях.

#### Монографії:

1. Основы теории и методы автоматизированного построения баз интеллектуальных имитационных моделей в СППР. - Киев: МВК МО, 1993. 116 с.
2. Технология композиционного построения имитационных моделей на базе Модифицированных сетей Петри. - Киев: МВК МО, 1992. 127 с. /У співавторстві/.

#### Статті та пріпріяти:

3. Метод повышения качества программы имитационных моделей. // Качество программных средств. - Калинин: НПО "Центр программистов", 1990. - С. 30 - 42. /У співавторстві/.
4. Анализ способов и средств описания процессов в программных системах имитационного моделирования // Методы, модели и системы решений задач прикладной математики. - Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 1995. - С. 95 - 104.
5. Программное обеспечение имитационного моделирования на ПЭЕМ // Системное моделирование - 17, Новосибирск, ВЦ СО АН СССР, 1991, с. 3-34. /У співавторстві/.

38

**АХАМОВ А. Г.**  
Основы теории и методы автоматизированного построения  
баз интеллектуальных имитационных моделей регулярного  
использования.

*Диссертация на соискание ученой степени доктора эконо-  
мических наук по специальности 08.03.03 - информационные  
системы в экономике. Киевский государственный экономический  
университет. Киев, 1995.*

Предложен новый подход, обеспечивающий автоматизацию  
процесса построения и использования баз интеллектуальных  
имитационных моделей регулярного использования на всем жиз-  
ненном цикле их существования. Информационная технология,  
поддерживающая данный подход, предполагает два основных эта-  
па: формирование метамоделей и генерацию программы на языке  
моделирования. Теоретически обоснована возможность и эффек-  
тивность применения Модифицированных сетей Петри при форми-  
ровании метамоделей. Предложена методика аттестации программ  
имитационных моделей. Спроектирована и разработана инструмен-  
тальная (программная) система ГИМН (Генератор Имитационных  
Моделей), поддерживающая технологию создания и сопровождения  
баз интеллектуальных имитационных моделей регулярного ис-  
пользования. Предложенные подходы проиллюстрированы рядом  
примеров использования данного класса моделей при создании  
внедренных СППР в частности, обеспечивающих управление круп-  
ным Банком и его филиалами, а так же на ряде предприятий и  
организаций Украины.

**Ключові слова:** інтелектуальні імітаційні моделі, генера-  
тор програм моделей, інформаційні технології.


**A. G. AHAMOV**

Fundamentals of Theory and Methods for Automatic Building  
of Base for Intelligent Simulation of Regular Application.

*Thesis for a Doctor's of Economics Degree in the  
speciality 08.03.03 - Information Systems in Economy. The  
Kiev State Economic University. Kiev, 1995.*

A new approach suggests automation of base building  
process and the use of bases for Intelligent Simulation of  
regular application during the period of its vital cycle.  
To support the approach the applied information technology  
proposes two main stages - the formation of a metamodel and  
the generation of a program in the Language of Modelling.  
The possibility and efficiency of application of Petri  
Modified Nets for the formation of a metamodel has been  
theoretically grounded. Methods of Simulation program  
attestation have been suggested. Instrumental (program) ISM  
(Generator of Simulation Models) System which supports the  
technology of creation and accompaniment of bases for  
Intelligent Simulation models of Regular Application has  
been designed and worked out. A number of examples  
illustrate the application the models of this class.  
Particularly such models have been used for creation and  
introduction of Decision Making Support Systems; they  
ensure for the management of a big bank and its branches and  
are applied at a number of enterprises in Ukraine.

**Key words:** Intelligence Simulation Models, Program  
Models Generator, Information Technologies, Base of  
Knowledge.



АВ 32.279

**АВ 32.279**

*[Faint, illegible text from the reverse side of the page, appearing as bleed-through.]*

Підписане до друку 4 05. 95 р.  
Формат 60x84 1/16, обсяг 7 с. п. л.  
Тир. 100 примірників в. зам. 7 с.  
Одеський державний економічний  
університет, відділ оперативної  
поліграфії 1,  
м. Одеса-100, вул. Преображенська, 8.