

УДК 681.3.06

В.Г. Левицький, аспір.
Житомирський інженерно-технологічний інститут

РОЗРОБКА МОВ ОПИСУ ЗАДАЧ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ В ПРОГРАМНОМУ КОМПЛЕКСІ “DSR OPEN LAB 1.0”

(Представлено к.т.н., доц. М.М. Колодницьким)

Запропонована методика побудови лінгвістичного забезпечення математичного моделювання, яка базується на розмежуванні функцій опису задачі між інтерфейсом і мовою та використанні графічного введення математичних формул. При цьому з проблемно-орієнтованих мов виділяється функціональність визначення виду обчислювальної задачі; завдання виду та умов аналізу математичної структури відбувається за допомогою графічного інтерфейсу програмної системи.

Програмна реалізація будь-якої нетривіальної проблеми потребує побудови інтерфейсу з користувачем, тобто, способу взаємодії людини та комп'ютера, який дозволяє в зручній формі формулювати завдання та переглядати результати роботи програми.

Дана робота присвячена вирішенню проблеми побудови інтерфейсу програмного комплексу (ПК) “DSR Open Lab 1.0” [1, 2] за рахунок розробки проблемно-орієнтованих мов опису задач математичного моделювання, які надають можливість простого, гнучкого та потужного опису різноманітних математичних структур.

Звернувшись до галузі математичного моделювання, виділимо два принципи побудови відповідного програмного забезпечення, альтернативних, з точки зору способу організації інтерфейсу людини та комп'ютера. Першим принципом є надання користувачу можливості використання певної універсальної мови програмування для вирішення проблем математичного моделювання. В цьому випадку користувач виступає одночасно в двох ролях, спочатку розробляючи прикладну програму для деякої порівняно вузької галузі, а потім і використовуючи даний програмний проблемно-орієнтований інструмент. Крайнім проявом такого принципу є повна відсутність підтримки конкретної галузі та формулювання задачі в певних універсальних термінах, скажімо, в термінах станів пам'яті комп'ютера. Прикладом цієї тенденції є застосування відомих мов програмування C, Pascal тощо.

Вдосконалення цієї процедури має характер застосування нових універсальних мов, які краще пристосовані для програмування задач конкретної предметної області, чи внесення змін до мов вже існуючих, розробка бібліотек програмування тощо. Серед останніх розробок прикладного програмного забезпечення згідно такого принципу, згадаємо, зокрема, приклади застосування мови програмування функціональної парадигми Haskell [3, 4] до побудови проблемно-орієнтованих програм.

Другий (з точки зору організації інтерфейсу) принцип побудови прикладного програмного забезпечення логічно впливає з попереднього та полягає в більш повному застосуванні мови програмування до потреб конкретної галузі. Це передбачає як позбавлення мови надлишкової універсальності, так і зосередження всіх її виразових засобів на проблемах розв'язку завдань предметної області, що в кінцевому результаті означає перетворення універсальної мови програмування на проблемно-орієнтовану мову. Зазвичай, створення лінгвістичного забезпечення такого роду супроводжується розробкою відповідних програмних середовищ, які, в свою чергу, є проблемно-орієнтованими та забезпечують підтримку всього циклу роботи користувача – в даному випадку, всіх стадій розв'язку задач математичного моделювання. Окремим випадком даного принципу є побудова прикладних програмних систем з жорстким інтерфейсом, які позбавляють користувача можливості гнучкої маніпуляції з визначенням задачі.

Недоліки обох крайнощів принципів побудови прикладного програмного забезпечення математичного моделювання є очевидними. В першому випадку користувач використовує інструментальний засіб, який занадто потужний та складний для його проблеми, що в загальному випадку суттєво ускладнює розв'язок конкретної задачі та примушує кожного разу вирішувати сторонні завдання – наприклад, програмувати використання системних ресурсів. В другому випадку надмірна жорсткість поведінки прикладного програмного забезпечення не

дозволяє вичерпно сформулювати проблему, яка потребує вирішення, а також позбавляє користувача можливості вирішення широкого кола проблем в рамках своєї предметної області.

Основні програмні засоби моделювання математичних задач [5–16] реалізують саме принцип використання проблемно-орієнтованих програмних середовищ і мов, схилившись при цьому, як було помічено в першому розділі, до надмірного ускладнення опису математичної задачі практично на всіх етапах: в частині лексики (велика кількість зарезервованих слів, зокрема, імен функцій), синтаксису (громіздкі конструкції, що запозичені з універсальних мов програмування) та семантики (складні принципи типізації тощо). І хоча бажання мати універсальний засіб визначення проблеми, незмінний для всього циклу вирішення математичної задачі, гнучкий та потужний, є цілком виправданим, якщо згадати широту завдань та функціональність програмних засобів моделювання математичних задач, та все ж застосування проблемно-орієнтованих мов, ускладнених до рівня мови програмування загального призначення (MATLAB, Mathematica®, Maple V тощо) є прийомом небажаним, а в певних випадках (наприклад, для використання в навчальному процесі) й абсолютно неприпустимим з огляду на специфіку кінцевого користувача, який може, взагалі кажучи, і не мати необхідних навичок програмування.

Тому, побудова проблемно-орієнтованих мов опису задач математичного моделювання ПК “DSR Open Lab 1.0” мала на меті модифікувати існуючі підходи до реалізації інтерфейсу людини та комп’ютера в сучасних прикладних системах, забезпечивши більш простий та прийнятний для непідготовленого користувача спосіб комунікації з програмним засобом за рахунок реалізації нової методики організації лінгвістичного забезпечення математичного моделювання – методики, що базується на розмежуванні функцій опису задачі між інтерфейсом і мовою та використанні графічного введення математичних формул.

Описані в даному розділі мови опису задач математичного моделювання є результатом поєднання двох способів розв’язання проблеми організації з користувачем – 1) застосування в програмній системі деякої проблемно-орієнтованої мови, як гнучкого інструмента опису задачі, та 2) створення візуальних, графічних інтерфейсів – на основі використання класифікації (типології) математичних моделей технічних систем, запропонованої в [17, 18]. У цих працях в основу типології математичних моделей покладено поняття математичних структур, з базового набору яких можуть будуватися більш складні моделі. Причому, з кожною математичною структурою в даному випадку пов’язується множина обчислювальних задач, яка визначається змістом математичної структури. (Так, наприклад, для обчислювальних задач математичної структури “Алгебраїчні системи з однією бінарною операцією – дискретні групи” список обчислювальних задач, власне, видів аналізу, буде вміщувати: операції з елементами групи – обчислення ланцюжка, побудова групи за генетичним кодом, вирахування потужності групи, знаходження всіх підгруп, розклад групи за підгрупами, знаходження нормальної підгрупи та фактор-групи).

Вказані в [17, 18] особливості типології математичних структур та її практична спрямованість дозволяє взяти її за основу побудови програмного комплексу математичного моделювання [2] та використати її як критерій розмежування функцій опису задачі між інтерфейсом та проблемно-орієнтованою мовою. Головною особливістю методики організації лінгвістичного забезпечення математичного моделювання, яка пропонується таким чином, стає винесення з проблемно-орієнтованих мов функціональності визначення виду обчислювальної задачі, необхідної користувачу, та завдання виду та умов аналізу певної математичної структури за допомогою графічного інтерфейсу програмної системи.

Згідно запропонованого підходу до організації взаємодії людина-комп’ютер в галузі математичного моделювання, проблемно-орієнтовані мови опису ПК “DSR Open Lab 1.0” є повністю декларативними та надають виключно засоби визначення певного математичного об’єкту, що, звичайно, суттєво полегшує вивчення та застосування ПК. Таким чином, запропонований поділ функцій між мовами опису задач математичного моделювання та рештою підсистем ПК “DSR Open Lab 1.0”, що реалізують графічний інтерфейс з користувачем, дозволяє реалізувати прості та зрозумілі лексику, синтаксис і семантику мов в формі, звичній для користувача, – загальноприйнятій математичній нотації.

Переваги перенесення функціональності, яка традиційно вважається частиною проблемно-орієнтованої мови, на інтерфейс можна проілюструвати рядом прикладів роботи ПК “DSR Open Lab 1.0” (рис. 1–6).

На рис. 1 показано, як за рахунок попереднього вибору математичної структури “Числові системи – натуральні числа” та виду аналізу “Великі числа” без введення надлишкового синтаксису вирішується проблема неоднозначності визначення константи натурального типу та бітового ланцюжка (великого натурального числа). При традиційному підході така колізія розв’язувалась би за допомогою конструкцій опису типу змінної.

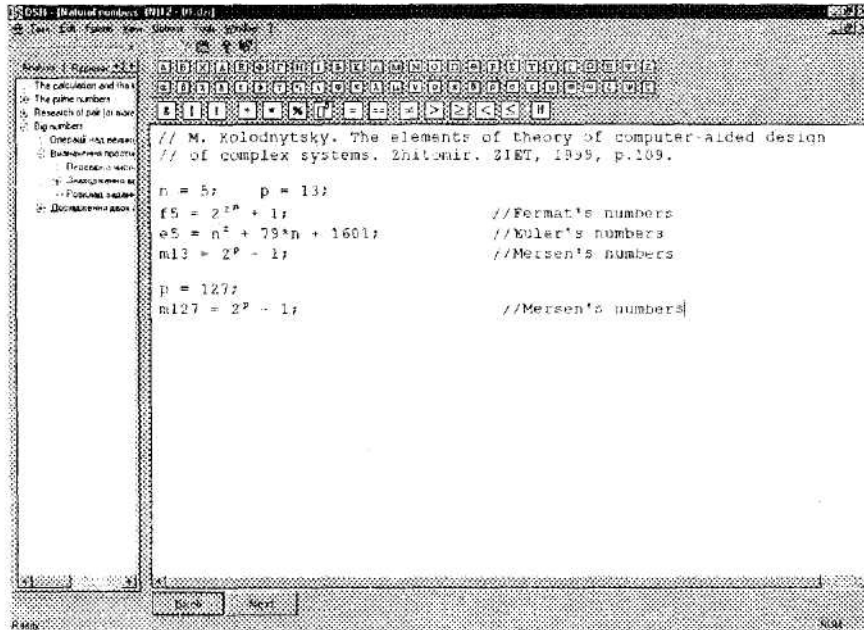


Рис. 1. Приклад визначення констант великих чисел

Збереження декларативності проблемно-орієнтованої мови за рахунок вибору видів аналізу та відповідних початкових умов за допомогою графічного інтерфейсу ілюструють рис. 2–6. Один і той самий опис математичної структури “Функціональні відношення” (рис. 2) дозволяє переглянути графіки функцій (рис. 3), таблицю значень функції на інтервалі (рис. 4), вирахувати визначений інтеграл (рис. 5) та дослідити властивості функцій (рис. 6) без застосування додаткових синтаксичних конструкцій для завдання виду аналізу чи, наприклад, інтервалів обрахунку функцій.

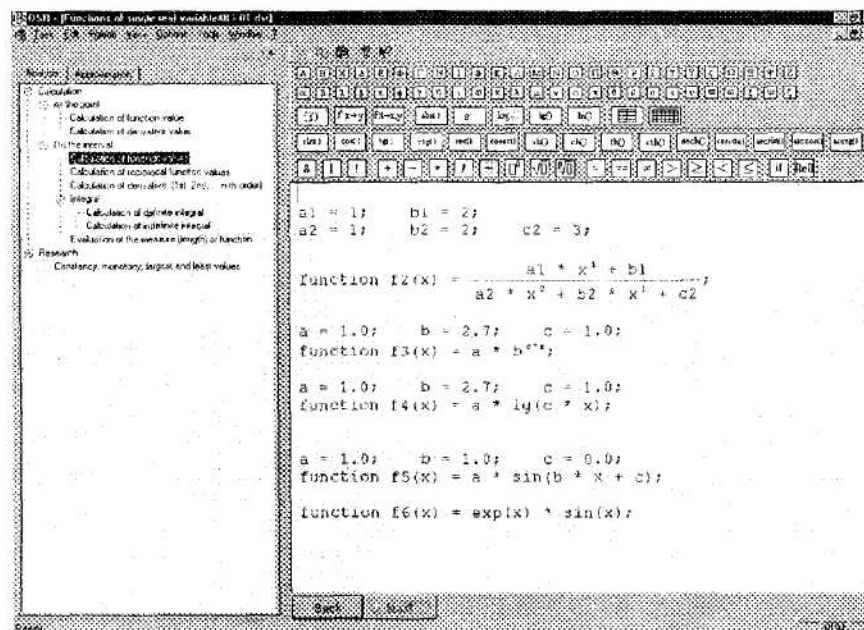


Рис. 2. Приклад визначення математичної структури “Функціональні відношення”

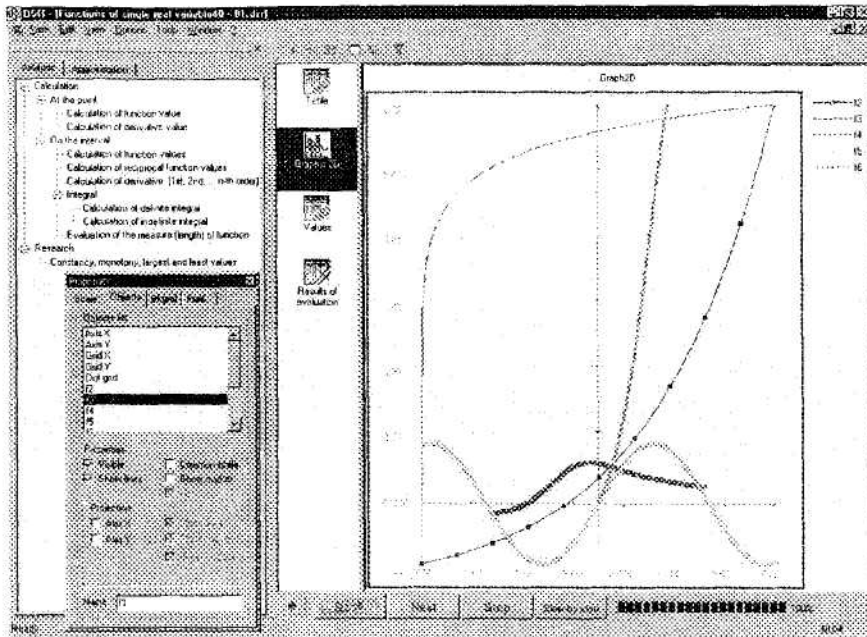


Рис. 3. Перегляд графіків функцій

X	f2
-3.400000	-0.166667
-2.969849	-0.164932
-2.938698	-0.163073
-2.909548	-0.161095
-2.879397	-0.158962
-2.849246	-0.156696
-2.819095	-0.154281
-2.788945	-0.151711
-2.758794	-0.148977
-2.728643	-0.146073
-2.698492	-0.142991
-2.668342	-0.139723
-2.638191	-0.136259
-2.608040	-0.132593
-2.577889	-0.128713
-2.547739	-0.124614
-2.517588	-0.120283
-2.487437	-0.115713
-2.457286	-0.110993

Рис. 4. Перегляд таблиці значень функції на інтервалі

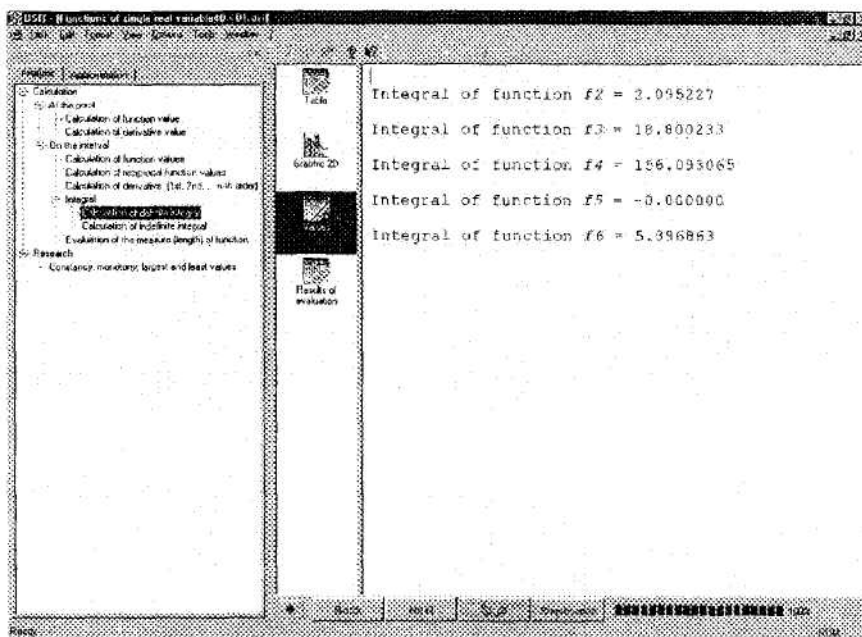


Рис. 5. Обрахування визначеного інтеграла

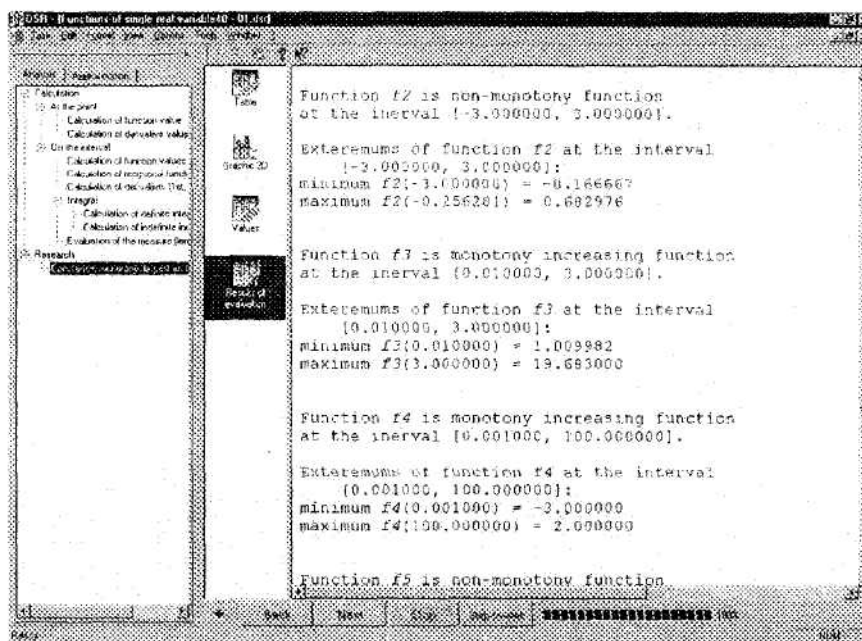


Рис. 6. Дослідження властивостей функцій

Іншою важливою особливістю методики організації лінгвістичного забезпечення математичного моделювання, яка пропонується, є реалізація інтегрованості ПК "DSR Open Lab 1.0", що розуміється в значенні наявності різноманітних допоміжних інструментів, однакових для різних класів задач, інструментів, які спрощують роботу за введенням початкових даних. Так ще одна частина функціональності, необхідної для визначення математичної моделі, перекладається з мов опису на інтерфейс програмної системи.

Зрозуміло, з одного боку, що, користування програмою не повинно викликати жодних труднощів у початківців. Інтерфейс повинен бути простим, в певній мірі стандартним, інтуїтивно зрозумілим для користувача. З іншого боку, він повинен бути достатньо потужним для вирішення складних задач, мати можливості опису даних мовою, що є максимально наближеною до предметної області, мати засоби для гнучкого управління процесом моделювання. Тому бажання забезпечити зручний інтерфейс з користувачем, не втративши при

цьому гнучкості та виразності вхідної мови, примушує ввести такий додатковий етап обробки вхідного опису задачі, який приховував би від користувача різноманітність конструкцій мови, перетворюючи математичний запис задачі до формату проблемно-орієнтованих мов опису задач математичного моделювання. Для цього підсистема лінгвістичного забезпечення ПК “DSR Open Lab 1.0” використовує модуль графічної побудови математичних виразів “DSR Equation Builder” – внутрішні структури даних “DSR Equation Builder” після конвертації стають вхідними даними для компіляторів проблемно-орієнтованих мов опису. Помітимо, що спосіб, за допомогою якого, з використанням спеціального синтаксису мови опису задач математичного моделювання, реалізують зв’язок із “DSR Equation Builder”, є повністю прихованим від користувача, що забезпечує простоту та наочність проблемно-орієнтованих мов.

Прикладом до вищесказаного може бути рис. 7, на якому зображений текст задачі аналізу раціональних чисел. Тут за рахунок графічної побудови констант раціонального типу розв’язується неоднозначність синтаксису між такими константами та виразами дійсного типу. При цьому текст задачі, який бачить користувач, відрізняється від того, що обробляється підсистемою лінгвістичного забезпечення, оскільки етап конвертації внутрішніх структур даних “DSR Equation Builder” формує опис, в якому відсутня неоднозначність.

```

// The calculation and transformation of arithmetic expressions
// vavilov V.V., Melnikov I.I., Olohnik S.N., Passichenko P.I.
// Mathematical tasks. Algebra. - Moscow, 1987, p.39 test#3

if 2/3 == 4/5
then
  result1 = 2/3;
else
  result1 = 4/5;
endif

if 6/7 > 24/28
then
  result2 = 6/7;
else
  result2 = 24/28;
endif

if 124/119 < 137/125
then
  result3 = 124/119;
else
  result3 = 137/125;
endif

```

Рис. 7. Приклад аналізу раціональних чисел

Отже, індивідуальні властивості опису кожної математичної моделі, врахування яких призвело б до переобтяження мов опису надмірно складними конструкціями, знаходять своє відображення в наборі кнопок на панелі інструментів “DSR Equation Builder”. Користувач звільняється від необхідності запам’ятовувати велику кількість імен функцій чи специфічний синтаксис мови. Натомість опис математичної структури зводиться до використання малої кількості простого виду речень мови та маніпулювання певним набором кнопок – опис задачі стає більш візуальним. Слід відзначити важливу особливість побудови панелей інструментів: для кожної математичної моделі користувач бачить мінімальний набір кнопок – виключно конструкції, необхідні для забезпечення повної функціональності опису задачі. Це дозволяє у випадку порівняно простих математичних структур, максимально спростити інтерфейс.

Розглянемо, наприклад, як змінюється інтерфейс підсистеми “DSR Equation Builder” при переході від аналізу натуральних чисел (рис. 8) до комплексних величин (рис. 9) – математична модель “Числові системи”.

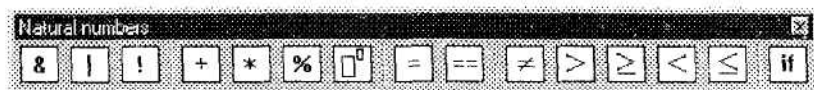


Рис. 8. Панель інструментів “Числові системи – натуральні числа”

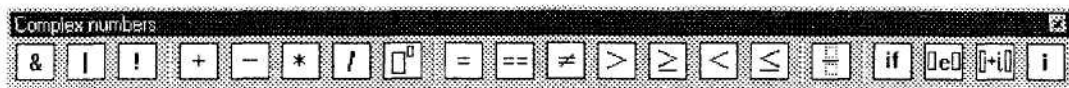


Рис. 9. Панель інструментів “Числові системи – комплексні числа”

Як бачимо з наведеного вище прикладу, навіть в рамках однієї математичної структури існує можливість суттєвого спрощення інтерфейсу користувача, що відіграє особливо важливу роль у випадку використання прикладної програмної системи в навчальному процесі. Користувач має змогу описувати довільні задачі обчислювального характеру, що передбачаються в рамках певної математичної моделі, і одночасно знає, що ніякі інші конструкції опису в даному випадку не допускаються. Більш того, всі конструкції постають перед користувачем, як загальноприйнятий математичний запис (наприклад, панель інструментів на рис. 10).

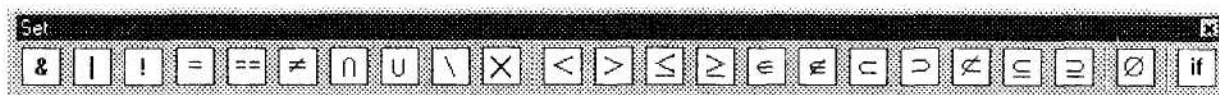


Рис. 10. Панель інструментів “Скінчені множини”

Крім того, використання “DSR Equation Builder” полегшує розробнику часткову оптимізацію коду, згенерованого за вхідним описом задачі. Наприклад, врахування серед кнопок підсистеми графічного введення математичних виразів специфічного вигляду матриці надає можливість більш ефективно організувати роботу із змінними відповідного типу, не переобтяжуючи в той же час, мови опису спеціальним синтаксисом.

Висновки

Викладені в даній роботі питання дозволяють зробити такі висновки:

1) Розробка проблемно-орієнтованих мов ПК “DSR Open Lab 1.0” модифікує існуючі підходи до реалізації інтерфейсу людини та комп’ютера в сучасних прикладних системах, пропонуючи таку методику побудови лінгвістичного забезпечення математичного моделювання, яка базується на розмежуванні функцій опису задачі між інтерфейсом і мовою та графічному введенні математичних формул.

2) Розмежування функцій опису задачі між графічним інтерфейсом користувача та підсистемою лінгвістичного забезпечення реалізовано на основі використання типології математичних моделей технічних систем, завдяки якій стає можливим винести з проблемно-орієнтованих мов функціональність визначення виду обчислювальної задачі, необхідної користувачу. Таким чином:

- а) завдання виду та умов аналізу певної математичної структури відбувається за допомогою графічного інтерфейсу програмної системи;
- б) проблемно-орієнтовані мови опису задач ММ є повністю декларативними та надають виключно засоби завдання певного математичного об’єкту;
- в) зникає необхідність ускладнення лексики проблемно-орієнтованих мов опису задач ММ великою кількістю зарезервованих слів (зокрема, іменами функцій вибору способу аналізу).

3) Підтримка інструменту графічної побудови математичних формул та автоматична конвертація результатів його роботи в речення мов опису задач ММ – на противагу до ручного кодування формул в текстовому вигляді самим користувачем – а) суттєво спрощують структуру підсистеми лінгвістичного забезпечення та б) дозволяють наблизити синтаксис визначення математичної структури до звичної математичної нотації.

ЛІТЕРАТУРА:

1. *Kolodnytsky M., Kovalchuk A., Kuryata S., Levitsky V.* The Mathematical Software Implementation for Computational Algebra and Number Theory // Proceedings of the 4th Asian Symposium on Computer Mathematics, December 17–21, Chiang Mai, Thailand, 2000. – P. 291–294.
2. *Колодницький М.М.* Тривимірна компонентна архітектура прикладної програмної системи “DSR Open Lab 1.0” як втілення концепцій реінженерії // Проблеми програмування. – 1998. – Вып. 4. – С. 37–45.
3. *Simon Peyton Jones, Mark Shields, John Launchbury, Andrew Tolmach.* Bridging the gulf: a common intermediate language for ML and Haskell. Proceedings of the 25th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, 1998. – P. 4.
4. *David J. King, John Launchbury.* Structuring depth-first search algorithms in Haskell. Papers of the 22nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages, 1995, Pages 344–354.
5. *Колодницький Н.М., Левіцький В.Г.* Обзор основных программных средств для моделирования математических задач // САПР и графика, 1999. – № 10 – С. 56–65.
6. *Потемкин В.Г.* Система MATLAB 5 для студентов. – М.: ДИАЛОГ-МИФИ, 1998. – 314 с.
7. *Потемкин В.Г.* Система MATLAB. Справочное пособие. – М.: ДИАЛОГ-МИФИ, 1998. – 350 с.
8. *John H. Mathews, Kurtis Fink.* Numerical Methods Using MATLAB. 3rd Edition, Pearson Education, 1999, pages 680.
9. MATLAB User's Manual, MATHWORKS Inc, MA, USA, 1999.
10. *Gaylord R.J., Kamin S.N. and Wellin P.R.* Programming with Mathematica. 2nd Edition, Springer-Verlag, TELOS, Santa Clara, 1996.
11. *Stephen Wolfram.* The Mathematica Book, Fourth Edition. Cambridge University Press, 1999, pages 1470.
12. MATHCAD 6.0 PLUS. Финансовые, инженерные и научные расчеты в среде Windows 95. / Пер. с англ. – М.: Информационно-издательский дом “Филинь”, 1996. – 712 с.
13. *Hecol K.M., Hansen M.L., Rickard K.M.* Maple V. Learning Guide. Springer-Verlag, 1996. – 274 p.
14. *Monagan M.B., Geddes K.O., Heal K.M., Labahn G., Vorkoetter S.M.* Maple V. Programming Guide. Springer-Verlag, 1996. – 383 p.
15. *Monagan M.B., Geddes K.O., Heal K.M., Labahn G., Vorkoetter S.M.* Maple V Release 5. Learning Guide. Springer-Verlag, 1998. – 284 p.
16. *Monagan M.B., Geddes K.O., Heal K.M., Labahn G., Vorkoetter S.M.* Maple V Release 5. Programming Guide. Springer-Verlag, 1998. 379 p.
17. *Колодницький М.М.* Типологія математичних моделей технічних систем. Частина 1 // Вісник ЖІТІ. – 1997. – № 6. – С. 130–142.
18. *Колодницький М.М.* Типологія математичних моделей технічних систем. Частина 2 // Вісник ЖІТІ. – 1998. – № 7. – С. 208–218.

ЛЕВИЦЬКИЙ В'ячеслав Георгійович – аспірант Житомирського інженерно-технологічного інституту.

Наукові інтереси:

- комп'ютерні інформаційні технології;
- моделювання і розв'язок задач за допомогою обчислювальної техніки;
- використання обчислювальної техніки в навчальному процесі;
- побудова компіляторів.

Подано 8.10.2000