

УДК 013.77:004.42; 37.013.03:004. 588(073)

М. Б. Головін – кандидат фізико-математичних наук,
доцент Волинського національного університету
імені Лесі Українки;

О. І. Сомик – аспірант Волинського національного
університету імені Лесі Українки

Психолого-педагогічні аспекти вивчення інформатики в процесі конструювання понятійних ієрархічних структур

Роботу виконано у ВНУ ім. Лесі Українки

У статті представлено інноваційну технологію автоматизованого навчання. Запропоновано формувати взаємозв'язані знання з інформатики в процесі оперування ієрархічними понятійними конструкціями. Описано прототип оригінальної програмної оболонки, що реалізує відповідний навчальний тренінг. Когнітивна (пізнавальна) психологія є концептуальною базою цієї роботи.

Ключові слова: програмне забезпечення, тестування, графи, інформаційні технології в навчанні.

Головін Н. Б., Сомик А. И. Психолого-педагогические аспекты изучения информатики в процессе конструирования понятийных иерархических структур. В статье представлена инновационная технология автоматизированного обучения. Предлагается формировать взаимосвязанные знания с информатики в процессе оперирования иерархическими понятийными конструкциями. Описан прототип оригинальной программной оболочки, которая реализует соответствующий учебный тренинг. Когнитивная (познавательная) психология есть концептуальным базисом этой работы.

Ключевые слова: программное обеспечение, тестирование, графы, информационные технологии в обучении.

Golovin N. B., Somik A. I. Psychological and Pedagogical Aspects of Computer Science Learning in the Process of Constructing Conceptual Hierarchical Structure. The innovative technology of automated studying is presented. It is proposed to form interconnected knowledge of science in the process of handling hierarchical conceptual structures. The prototype of original program environment, which implements the relevant educational training, is described. Cognitive (discovery) psychology is the conceptual basis of this work.

Key words: software testing, graphs, information technology in education.

Постановка наукової проблеми та її значення. Актуальною проблемою, на думку авторів, є розробка програмних засобів для підтримки навчальних процесів, у яких відбувається не тільки просте понятійне відтворення, а й реалізується навчальне причинно-наслідкове, абстрактно-логічне масштабування понятійними одиницями. Останнє може досягатися в процесі конструювання зв'язаних понятійних ієрархій.

Метою цієї роботи є розгляд підходів до створення ефективного тренажерного програмного засобу, що оперує завданнями на конструювання ієрархічних структур.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Виклад оригінальної частини основного матеріалу в цій статті здійснено нижче в процесі розгляду конструювання навчальних понятійних ієрархій та аналізу програмних засобів і завдань, необхідних для цього.

1. Особливості автоматизованого навчання

Основою автоматизованої діагностики процесу навчання в комп'ютерному класі є добрий диференційований зворотний зв'язок між викладачем та групою суб'єктів навчання, які виконують серії завдань у межах відповідних тренажерних програм. Цей зв'язок не безпосередній, а опосередкований; він реалізується через комп'ютери, пов'язані в локальну мережу. Останнє дає змогу значною мірою автоматизувати та індивідуалізувати процес навчання.

Ефективність автоматизованого навчального процесу значною мірою залежить від програмних засобів, які для цього використовуються, та структури завдань. Дієвість пакета завдань визначається інформаційним наповненням тренажерної програми: змістом завдань, їх структурою та послідовністю пред'явлення. Пакети завдань, однорідні за кількісними та якісними показниками, дають змогу значно спростити механізм керування навчанням. У цьому сенсі важлива формалізація понять «кількість» і «складність» у розумовій роботі щодо кожного завдання.

Отримання кількісних параметрів будь-якого процесу, у тому числі пізнавального, передбачає розуміння закономірностей його протікання. У когнітивно-орієнтованих течіях теоретичної психології активно розвивається ідея про те, що інтелектуальна діяльність значною мірою детермінується структурною організацією пізнавальної сфери [1].

Однак реальний типовий, тестологічний підхід у навчанні, як правило, акцентований не на психологічному механізмі процесу навчання, а на вибірковій перевірці засвоєння декларативних знань. Такий підхід дає мало користі для формування спеціалістів у сфері високих технологій.

2. Специфіка розумових навчальних дій під час програмування.

Практична діяльність у технологічній галузі щодо складних, добре формалізованих штучних об'єктів пов'язана з поетапним абстрактно-логічним, причинно-наслідковим, продуктивним мисленням, яке детермінує цю діяльність. Це стосується діяльності зі створення, модернізації та відлагодження складноорганізованих об'єктів, наприклад таких, як програмні засоби й електронні пристрої. Діяльність у цій сфері первинна. Вона формує процеси сприйняття; спосіб диференціації та структурування знань; особливості зв'язків у понятійній сфері; методологію дій – і розумових, і матеріалізованих.

Специфіка практичної навчальної діяльності щодо складних, добре формалізованих штучних об'єктів полягає в тому, що людина не може оперувати одночасно всіма численними функціональними вузлами складного об'єкта, які вона сприймає й розрізняє. Їх число навіть у навчальних завданнях значно перевищує ту кількість, яку можна одночасно утримувати в полі уваги. Суб'єкт навчання змушений у процесі інтелектуальної роботи масштабувати поняттями, не виходячи в процесі ментальних дій, що усвідомлюються, за межі магічного числа Міллера 7 ± 2 [2], котре характеризує обсяг короточасної пам'яті людини. У результаті тривалої інтелектуальної діяльності, суть якої – абстрактно-логічне масштабування поняттями, у довготривалій декларативній пам'яті поступово формується ментальна вербальна конструкція ієрархічного типу – пізнавальна (когнітивна) схема об'єкта навчання (наприклад програми). Особливості навчальних дій щодо програмних об'єктів у контексті ієрархічних ментальних структур розглянуто в роботі [3].

Методологія професійної діяльності у сфері програмування ґрунтується на відомих методах низхідної поетапної деталізації та модульного програмування [4, 5]. Застосування цих методів опосередковано підтверджує описаний вище механізм мислення, який спирається на ієрархічну конструкцію. Так, при низхідній поетапній деталізації ієрархічна конструкція вибудовується зверху вниз, тобто від поодиноких узагальнених понятійних одиниць до великої кількості конкретизованих одиниць – операторів мови програмування. Саме тоді коли кожен окремий крок вирішення деталізованої проблеми можна представити окремим оператором, і починається процес безпосереднього програмування. Технологія модульного програмування сповідує альтернативний спосіб мислення: від великої кількості конкретизованих одиниць операторів до окремих узагальнених, що характеризують програмні блоки. По суті, остання технологія дає змогу представити в мисленні велику сукупність логічно зв'язаних елементів як монолітну, поодинокую відокремлену сутність, що характеризується окремим поняттям або короткою концепцією.

3. Конструювання понятійних ієрархій як навчальний захід.

У тренажерних програмних засобах використовували завдання двох типів.

У першому типі завдань порядок слідування однорівневих елементів ієрархічної структури не мав значення. У процесі апробації цих завдань використано структуру обчислювальної системи, схему типів файлів у файловій системі Windows, дерево типів цих мови програмування Delphi.

У другому типі завдань порядок слідування однорівневих елементів ієрархічної структури мав значення. Апробація цих завдань реалізовувалася під час конструювання алгоритмів та програм у вигляді понятійних ієрархій.

Оператори програми в завданнях другого типу є найбільш конкретними елементами ієрархії. Узагальнені понятійні одиниці складають надбудову над операторами. Вони об'єднують сукупності логічно зв'язаних операторів. Ключова відмінність завдань другого виду від першого полягає в тому, що під час перевірки виконання цих завдань відстежується не тільки наявність окремих гілок дерева, а і їх порядок. Однорівневі елементи структури є окремими кроками у розв'язання поставленого завдання.

Ілюстрація формалізації об'єктів навчальної діяльності за допомогою програмних деревоподібних структур є цікавою, адже програма як навчальний об'єкт легко масштабується, за розміром і складністю вона добре формалізована й структурована. Діяльність учня в процесі навчального

програмування відбувається у вікні текстового редактора та завжди доступна для автоматизованого моніторингу за допомогою діагностичної програми, що працює паралельно із засобами програмування в багатозадачній операційній системі Windows. Доступний також і результат дії навчальної програми, щойно створеної, виправленої або модернізованої учнем.

4. Прототип автентичної програми для начальних дій з ієрархіями.

Розглянемо інтерфейс програмного забезпечення, розробленого спеціально для апробації ідей покращення ефективності навчання завдяки застосуванню завдань на конструювання понятійних ієрархій. На рисунку 1 показано вікно виконання завдання першого типу. Тут потрібно побудувати структуру обчислювальної системи з наявних елементів.

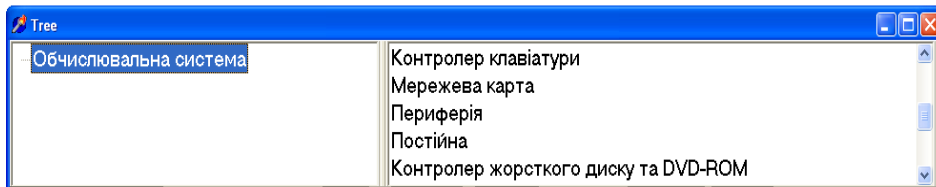


Рис. 1. Інтерфейс початкової програми, що оперує завданням першого типу.
Стартова ситуація під час виконання завдання

Запропонований студентам інтерфейс вікна виконання складається з двох частин (рис. 1). Із лівого боку розміщене вікно, у якому потрібно побудувати ієрархію, а з правого – вікно з компонентами для конструювання. На початковому етапі розв'язання завдання панель зліва містить тільки кореневий елемент дерева, котрий задає предметну область. Виконання завдання полягає в побудові структури способом переміщення компонентів справа наліво. Порядок слідування елементів у списку справа випадковий.

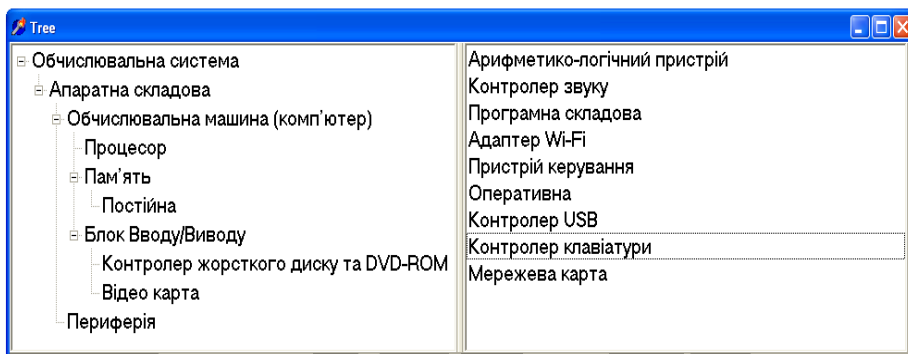


Рис. 2. Інтерфейс початкової програми, що оперує завданням першого типу.
Завдання в процесі виконання

На рисунку 2 показано ситуацію, коли завдання на конструювання виконано ще не повністю, тільки частину компонентів включено в ієрархію.

У деяких випадках список справа може бути й надлишковим. У такому разі наприкінці правильного виконання завдання права панель усе ж таки містить деяку кількість записів. Такі завдання надають побудові структури додаткової складності.

Розглянемо другий тип завдань. Програмну структуру виконано у форматі ієрархії, схожій на структуру класифікації понятійних одиниць. Однак у випадку програми зміст окремих елементів конструкції дещо інший. Вузли графа – це концепції або поняття. Вони об'єднують у єдине ціле операторів мови програмування. У такій конструкції важливий порядок елементів.

Приклад завдання представлено на рисунку 3. Пропонується сконструювати програму, яка відокремлює окремі слова в реченні. Це завдання може бути розв'язане, якщо надати змінним початкових значень, реалізувати однократну вирізку слова, здійснити багатократну вирізку окремих слів, організувати вивід результату. Існують і менші проблеми, які слід вирішити при цьому, а саме: пошук поточного пробілу, перетворення кінця чергового слова на початок наступного.

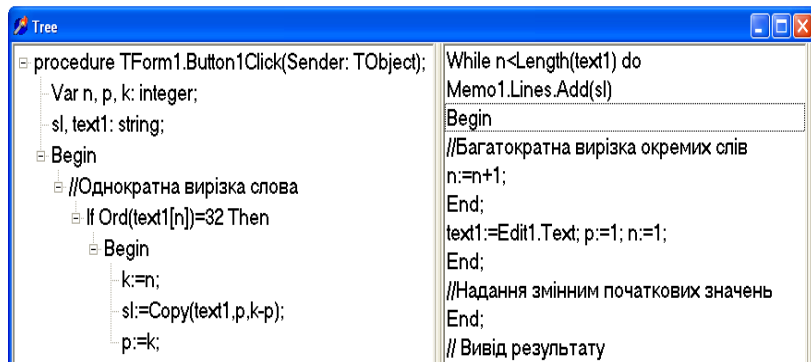


Рис. 3. Інтерфейс початкової програми, що оперує завданням другого типу. Завдання виконано не повністю

На рисунку 3 показано проміжний стан у розв’язку задачі. Частина компонентів, що були розміщені в панелі з правого боку на початку виконання завдання, уже включені в дерево. Залишилося визначити місце й перетягнути наліво компоненти, які ще залишилися справа.

Суттєвим є те, що порядок компонентів правої панелі випадковий.

Змінною Text 1 позначено початковий текст. Змінна N відстежує номер поточної букви при пошуку чергового пробілу в тексті. Змінними P і K позначено, відповідно, положення пробілу на початку й наприкінці слова. Змінна Sl містить поточне вирізане слово.

Зауважимо, що в процесі конструювання інтерфейсу програми автори намагалися зробити маніпуляції в ньому максимально швидкими. Редагування текстів у цьому інтерфейсі мінімізоване. Роботу реалізовано переважно за допомогою миші. Крім перенесень окремих компонентів між деревом і списком, доступні також перестановки як усередині списку, так і в межах дерева, реалізовано й можливість перенесення цілих гілок дерева.

Максимальне пришвидшення маніпуляцій упродовж виконання завдання, на думку авторів, – ключовий принциповий момент, що дає змогу акцентуватися на інтелектуальних діях, а не на рутинній психомоторній роботі редагування текстів. Цей підхід дає змогу оцінювати швидкість інтелектуальних дій. Психомоторні дії, пов’язані з уведенням відповіді, мінімізовані. Аналіз структури деревоподібної структури, що є результатом виконання завдання, дає змогу визначити, який відсоток гілок дерева правильний. Така дискретизація під час перевірки більш тонко описує процес освоєння навчального матеріалу, ніж перевірка, що передбачає два значення – правильно й неправильно. На рисунку 4 показано схему роботи програми.

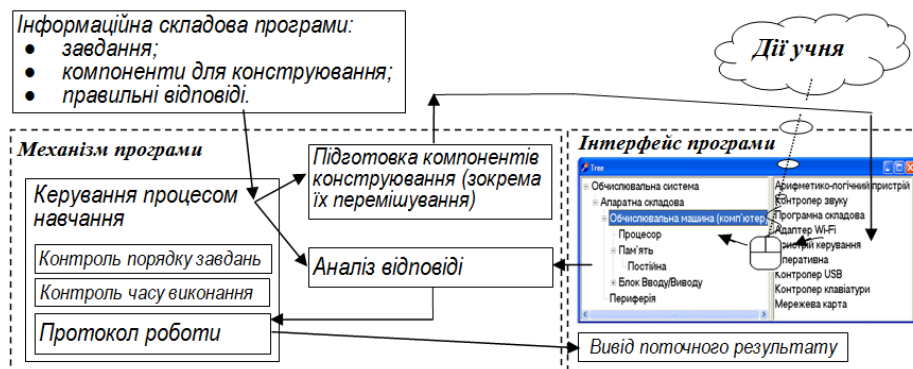


Рис. 4. Механізм роботи програми в загальних рисах

5. Однорідність пакетів завдань на побудову ієрархії.

Важливим моментом, який впливає на адекватну діагностику навчальних комп’ютеризованих процесів, є однорідність відповідних пакетів завдань.

Однорідність завдань у пакеті можна оцінити за двома факторами – за кількістю дій, необхідних для виконання кожного із завдань, та за складністю окремих завдань у пакеті. Зрозуміло, що в однорідному пакеті кожне завдання має бути каліброване за зазначеними факторами.

Розглянемо способи досягнення однорідності пакетів завдань для випадку, коли вони укомплектовані саме завданнями на конструювання ієрархій.

Калібровка за кількістю дій реалізується легко, оскільки кількість дій корелює з кількістю елементів конструювання. Для досягнення однорідності за кількістю дій достатньо запроєктувати ці завдання так, щоб кількість елементів конструювання в різних завданнях була однаковою.

Ієрархічні структури добре масштабуються. Базова для формування завдання ієрархічна структура, як правило, значно більша, ніж потрібно для створення одного завдання, і включає елементи різного ступеня узагальненості. Тому під час проектування завдань на конструювання гілки цих деревоподібних утворень можна згортати й розгортати на власний розсуд, не втрачаючи змістовної цілісності, але регулюючи при цьому кількість елементів у структурі.

Інший цінний момент полягає в тому, що окрема гілка дерева є теж ієрархією. Тому як базу для завдання можна вибрати не деталізацію кореня всього дерева, а деталізацію окремої гілки. Це дає змогу досягнути найвищого ступеня деталізації та конкретизації матеріалу, не втрачаючи при цьому однорідності завдання за кількістю дій, необхідних для виконання завдання.

Великий розмір базової для завдань ієрархії дає можливість утворювати десятки завдань навколо однієї деревоподібної структури, масштабуючи її при формуванні завдання або пересуваючи. Якщо інтерпретувати програму як ієрархію, останнє твердження теж дійсне. Тут, з одного боку, можна акцентувати увагу при формуванні завдання на виготовленні окремих процедур, відповідних локальним гілкам дерева; з іншого – використання готових бібліотек процедур може дати змогу пересунути ближче до «стовбура» ієрархії, базової для завдань програми.

Калібрування складності теж непогано формалізується для завдань на конструювання ієрархічних конструкцій. Повтори та розгалуження в тексті програми, а також складні математичні вирази, ексклюзивні функції й процедури (підпрограми) користувача підвищують складність програми. Такі фрагменти програми жорстко задають межі уваги під час їх обмірковування.

Кожен із перерахованих вище фрагментів формує свій змістовний складник, а значить спричинює локалізацію уваги на собі, ініціює щодо себе відповідні логічні дії, формує нову цілісну узагальнену понятійну одиницю або коротке концептуальне твердження. Зрозуміло, що складність розумових висновків збільшується в тих випадках, коли структура має вкладення. Понятійну одиницю верхнього рівня не можна отримати без концептуальної одиниці нижнього. Як видно із прикладу, із написанням коду програми, що реалізує вилучення слів із речення, ми не можемо отримати багатократну вирізка слів без однократної вирізки, а однократна вирізка не може бути реалізована без пошуку чергового пробілу.

У програмах також існують фрагменти тексту, достатньо прості для сприйняття й створення. Це, зокрема, лінійні частини програми, порядок виконання окремих кроків яких є незмінним. Ускладненими можна вважати лінійні фрагменти, які сприймаються тільки сукупно, у певному комплекті складових частин. Окрема частина такого фрагмента має значення тільки в контексті загальної мети роботи програми.

Узагальнюючи наведене вище, вважатимемо, що міра вкладеності ієрархічних програмних конструкцій, використаних для створення завдання, є мірою складності. Детальніше це розглянуто в роботі [6].

Висновки та перспективи подальших досліджень. Отже, конструювання ієрархічних понятійних структур є ефективним, оригінальним та перспективним підходом для подальшого розвитку в реалізації автоматизованого навчального тренінгу та тестування.

Навчальні програми й завдання, що підтримують роботу з ієрархічними конструкціями, активують і підтримують у процесі навчання розвиток причинно-наслідкового та абстрактно-логічного мислення.

Витрати часу на матеріалізовану діяльність у завданнях на конструювання ієрархічних структур мінімізуються порівняно зі звичайними завданнями, зменшується дифузність мислення, підвищується диференціація понятійної бази розумових дій.

Важливим моментом для досліджень у галузі психології й педагогіки навчання, а також для автоматизації навчальних процесів є те, що пакети завдань на конструювання ієрархічних структур можна зробити однорідними за складністю та кількістю матеріалізованих дій.

Список використаної літератури

1. Холодная М. А. Психология интеллекта: парадоксы исследования / Холодная М. А. – СПб. : Питер, 2002. – 272с.
1. Андерсон Дж. Когнитивная психология / Дж. Андерсон. – 5-е изд. – СПб. : Питер, 2002. – 496 с.

2. Головін М. Б. Зміст підготовки висококваліфікованого фахівця з інформаційних комп'ютерних технологій у контексті когнітивних процесів (на прикладі програмування) / М. Б. Головін // Інформаційні технології в освіті. – Херсон, 2008. – Вип. 2. – С. 66–73.
3. Дал У. Структурное программирование / У. Дал, Э. Дейкстра, К. Хоар. – М. : Мир, 1975. – 246 с.
4. Хьюз Дж. Структурный подход к программированию / Дж. Хьюз, Дж. Митчом. – М. : Мир, 1980. – 276 с.
5. Головін М. Б. Кількість і складність розумових дій у контексті діагностики когнітивних процесів, що детермінують практику навчального програмування / М. Б. Головін // Вісн. Черкас. ун-ту. Серія : Пед. науки. – Черкаси, 2008. – Вип. 125. – С. 34–41.

Статтю подано до редколегії
16.12.2011 р.