

АНАЛІЗ ТА ОПТИМІЗАЦІЯ ЦИФРОВИХ І АНАЛОГОВИХ СИСТЕМ

УДК 004.514.62+621.382.33

М. Близнюк, О. Маркелов

Національний університет "Львівська політехніка",
кафедра систем автоматизованого проектування

АНАЛІЗ ПІДХОДІВ ДО РОЗРОБКИ ДІАЛОГОВИХ СИСТЕМ ТА ЇХ РЕАЛІЗАЦІЯ В ІНТЕРАКТИВНОМУ ГРАФІЧНОМУ РЕДАКТОРІ СХЕМ СИСТЕМИ "Micro-PC"

© Близнюк М., Маркелов О., 2001

Обґрунтовано необхідність подальшого розвитку діалогових систем (ДС) у розробці САПР. Проведено системний аналіз ДС відносно класифікації за різними ознаками, систематизації типів, форм та режимів діалогу, аналізу метафор та позитивних якостей інтерактивної графіки та їх впливу на ефективність роботи користувача. Показано застосування системно проаналізованих особливостей та принципів розробки ДС на прикладі графічного редактора електричних схем "Schematic 2000" для навчально-дослідної системи схемотехнічного проектування "Micro-PC".

1. Мотивація необхідності використання сукупності діалогових форм при реалізації систем та підсистем автоматизованого проектування

1.1. Для чого необхідні діалогові системи

Використання діалогу – оперативної взаємодії користувача з комп'ютером під час розв'язання задачі – відіграє значну роль в автоматизації проектування. Режим діалогу дає змогу розширити коло задач, що вирішується системами автоматизованого проектування (САПР), прискорити розв'язання, виключити проміжних виконавців між проектувальником та комп'ютером. Особливо зручний графічний діалог-обмін інформацією між користувачем та комп'ютером у звичній для проектувальника формі: у вигляді креслень, графіків, схем.

У режимі такого діалогу користувач та комп'ютер можуть по чергові вирішувати низку взаємопов'язаних задач, створюючи в пам'яті машини інформаційну модель проектного об'єкта. Досвід та інтуїція проектувальника використовується для розв'язання творчих, важко формалізованих задач, а можливості комп'ютера – для складних розрахунків, рутинних операцій, які характеризуються підвищеною ймовірністю помилок.

Крім набору програм прикладних розрахунків, діалогова система містить програму керування діалогом, програму побудови зображення, інтерпретатор повідомлень користувача [1].

Збільшення кількості задач, що вирішуються в діалозі, їхня складність, підвищення вимог до своєчасності, достовірності, до повноти та зручності інформації, що подається,

зумовлює необхідність подальшого удосконалення методології проектування систем із використанням діалогових підсистем, яка повинна враховувати не тільки особливості людського фактора, але й вимоги до забезпечення максимальної ефективності технічного, програмного та інформаційного забезпечення, що використовується.

Розробка ефективних діалогових підсистем є комплексною проблемою, яка містить аналіз і типізацію інформаційних вимог користувачів, синтез типової моделі діалогу для заданої множини користувачів, інформаційні запити яких належать одній предметній галузі, синтез інформаційного та прикладного програмного забезпечення, оптимального за заданими критеріями ефективності [2].

Неможливість повної формалізації проектування викликає необхідність участі проєктувальника у розробленні радіоелектронної апаратури (РЕА) з використанням діалогових засобів САПР. Інтерактивний (діалоговий) вплив з боку користувача на отримання проєктного рішення є важливим фактором забезпечення єдності та взаємозв'язку етапів структурного, функціонального, схемотехнічного та конструкторського проектування РЕА [3].

Незважаючи на інтуїтивну схильність розробників прикладних програмних систем до використання діалогових форм, розроблення ефективних діалогових систем потребує системного аналізу (систематизації, класифікації, декомпозиції, узагальнення підходів тощо) наявного матеріалу, накопиченого на основі досвіду проєктувальників, що розробляли діалогові системи. Об'єктом системного аналізу в даному випадку повинна служити *діалогова система (ДС)* як сукупність технічних, програмних та інформаційних засобів, що забезпечують задане коло користувачів можливістю розв'язання деякої множини задач в інтерактивному режимі взаємодії з комп'ютером відповідно до сценаріїв діалогу. Системному аналізу підлягають такі сторони об'єкта ДС, як: *класифікація за різними ознаками (типом користувачів, проблемною орієнтацією, методами організації діалогу та програмного забезпечення); систематизація та класифікація типів, форм та режимів діалогу; аналіз метафор, що покладені в основу діалогу та аналіз позитивних якостей інтерактивної графіки і їх вплив на ефективність роботи користувача.*

1.2. Класифікація діалогових систем

Класифікують ДС за такими ознаками: 1. За типом користувачів розрізняють ДС, які взаємодіють: а) із випадковими користувачами, взаємодія яких із базою даних ДС не зумовлена їхньою постійною роботою і потреби яких у діалозі непередбачені; б) із параметричними користувачами, взаємодія яких з базою даних ДС є невід'ємною складовою частиною роботи, яка виконується щодня і регламентована чіткими правилами взаємодії з ДС; в) із дослідниками та розробниками систем, взаємодія яких із базою даних ДС є частиною їхньої роботи, але потреби яких в діалозі не завжди непередбачені; г) із прикладними та системними програмістами. 2. За проблемною орієнтацією розрізняють такі типи ДС: а) прийняття рішень при плануванні та оперативному управлінні функціонування об'єктів різного класу; б) інформаційного пошуку та навчання; в) автоматизації проектування, орієнтовані на розробку технічних завдань, технічних та робочих проєктів, креслень, технологічної документації; г) автоматизація наукових досліджень, орієнтованих на проведення та оцінку результатів експериментів на реальних об'єктах чи їхніх моделях; д) управління процесом діалогу й виконання сервісних функцій. ДС цього типу виконують такі функції: організація діалогу (початок сеансу діалогу,

переривання, перемикання типів діалогів); керування користувачем послідовністю обробки задач; ввід-вивід даних, редагування, розміщення інформації; організація та виконання сервісних функцій, функцій допомоги користувачу в складних ситуаціях; зв'язок користувачів системи один з одним. 3. За методами організації діалогу розрізняють такі типи ДС: *а)* діалоги, що реалізуються за допомогою директив (команд); *б)* меню; *в)* заповнення бланків (діалог за шаблоном); *г)* альтернативний чи бінарний діалог; *д)* природна мова. 4. За методами організації програмного забезпечення розрізняють такі типи ДС: *а)* жорсткого типу, в якому формування запиту проходить за фіксованим набором заздалегідь описаних схем; *б)* гнучкого типу (адаптивні), в яких формування запиту можливе за різними схемами (до них належать системи, які використовують лінгвістичний процесор, що здійснює морфологічний та синтаксичний аналізи сформованого запиту [2]).

1.3. Типи та форми діалогу

Тип діалогу характеризує структурні та лінгвістичні засоби для оформлення діалогових обмінів: *а) абстрактний тип* визначає основні відмінності та властивості діалогового обміну (на цьому рівні описуються такі основні компоненти діалогового обміну, як дії і реакції на нього, а також їхня взаємодія з позицій: характер інтерпретатора і дії; можливості вибору; однозначності формулювання задачі); *б) конкретний тип* діалогу пов'язаний із характеристикою повідомлень, що входять у діалоговий обмін. Розглядаються такі складові: алфавіт і словник (елементи представлення повідомлень); структура представлення повідомлень (внутрішні та зовнішні формати); структура вмісту повідомлення (семантичні властивості, форма, інтерпретація). В табл.1 зведено порівняльну характеристику основних типів діалогу.

Таблиця 1

Порівняльна характеристика типів діалогу

Абстрактний тип діалогу	Конкретний тип діалогу	Властивості діалогу		
		Характер ініціатора та дії	Можливість вибору функції	Однозначність формулювання
Простий запит	Ввід даних Корекція даних Перегляд даних Отримання підказки	Система, пропозиція	Фіксована	Однозначне
Пропозиція для вибору	Меню функцій Меню параметрів Запитання "Так/ Ні"	Система, пропозиція	Обмежений	Однозначне
Запит з синтаксисом відповіді	Ввід даних за форматом	Система, запит	Обмежений	Однозначне
Запит для вільної відповіді	Розповідне речення	Система, запит	Необмежений	Неоднозначне
Директива (команда)	Команда, Запит допомоги	Користувач, запит	Необмежений	Однозначне
Фраза на обмеженій природній мові	Питальне речення, Розповідне речення	Користувач, запит	Необмежений	Неоднозначне

Форма діалогу характеризує вміст діалогового спілкування, описує повторення однакових типів діалогу та поєднання різних типів в одному обміні.

Директивна форма діалогу характеризується тим, що при виклику виконання операції застосовується директива (команда). Ініціатором обміну є користувач; можливості вибору операцій необмежені; інтерпретація запиту однозначна та виконується автоматично. Таблична форма діалогу містить такі види діалогу: вибір операції для використання з меню; заповнення та редагування бланка (шаблону) даних; вивід результатів у табличній формі. Фразова форма діалогу використовує обмежену природну мову. Її складають розповідні, питальні, окличні речення. Використовується обмежений словник та послідовність слів. Об'єктно-орієнтований діалог є аналогією людини у світі об'єктів у поєднанні з принципом безпосереднього спілкування. Робота користувача на комп'ютері полягає в цілеспрямованій зміні деякого об'єкта, який має внутрішню структуру, певний вміст та зовнішнє символічне чи графічне представлення. Об'єкт розуміють в широкому змісті слова. Користувач має можливість створювати світ об'єктів, використовуючи його, впливати безпосередньо на окремі об'єкти, змінюючи їхні атрибути та зв'язки з іншими об'єктами.

1.4. Діалогові режими

Досвід створення та експлуатації діалогових систем як загального, так і спеціального призначення свідчить про відмінності підходів до проектування структури діалогової взаємодії. Ці відмінності виявляються, насамперед, у тому, кому віддається ініціатива з управління діалогом та відображають його орієнтацію на певний рівень підготовки користувача навіть у межах одного й того ж класу задач. Залежно від того, кому належить керуюча функція, виділяють такі режими ведення діалогу: *а) режим “ведучий – комп'ютер”*, в якому діалог спрямовується запитом з боку системи. За користувачем залишається право вибору однієї з пропонованих системою альтернатив – прийняття чи відхилення запропонованих можливостей, чи вводу вхідних даних. Інформація з боку комп'ютера у цьому випадку видається в повному обсязі з максимальною деталізацією. Для досвідченого користувача режим “максимальної підказки” виявляється непотрібним і, більш того, небажаним, оскільки для нього найкращий або діалог з “двостороннім керуванням”, або максимально “стилий” діалог із переважним керуванням з боку людини, що є зовсім неприпустимим для недосвідченого користувача; *б) режим “ведучий – користувач”*, в якому діалог спрямований командами з боку користувача (на комп'ютер у такому режимі покладаються функції виконання запитів користувача); *в) режим “рівноправних партнерів”*, в якому функція керування діалогом переходить від користувача до комп'ютера і навпаки, залежно від деяких формальних показників якості функціонування системи людина-машина, до яких належать: статус користувача в системі; тривалість перерви в роботі; кількість помилок; кількість звертань до інструкцій; ступінь відповідальності виконання даного кроку.

Переходять з одного режиму в інший або на вимогу користувача, або на основі оцінки ситуації під час спілкування, яка складається [3].

1.5. Інтелектуалізація спілкування проектувальника із системою

Під час організації діалогу розвивається тенденція інтелектуалізації інтерфейсу користувач-комп'ютер. “Дружність” інтерфейсу забезпечується імітацією звичної для користувача роботи з об'єктом, основаної на максимальному використанні технічних можливостей персональних комп'ютерів. Старі методи роботи, що реалізуються новими засобами, набувають метафоричного характеру.

При інтелектуалізації спілкування проектувальників САПР можуть використовуватися такі прийоми та принципи: *а) метафора письмового стола – імітація на екрані поверхні письмового стола з розміщеними на ньому документами* (при цьому застосовується так зване багатовіконне відображення інформації: екран ділиться на прямокутні чи іншої геометричної форми вікна, які можуть перекриватися, у кожному вікні відтворюється фізичний образ документа чи його частини, а засоби керування курсором, “миша” та клавіатура дають змогу знаходити, переглядати, редагувати, переміщувати чи знищувати документи на екрані); *б) метафора великоформатного бланка – розрахунок і представлення результатів у вигляді двовимірної електронної таблиці* (кожна комірка таблиці має певне значення і може бути зв’язана з розрахунковою формулою, і при внесенні змін у будь-якій комірці таблиці автоматично перераховуються усі залежні від неї величини в інших комірках); *в) принцип безпосереднього спілкування – оснований на максимальному відштовхуванні від мовного спілкування (директивного й навіть природно-мовного) на користь безпосередньої безмовної взаємодії користувачів із системою*; робота користувача за дисплеєм має маніпулятивний характер, (“дивитися та діяти”, а не запам’ятовувати та набирати текст); для виконання маніпуляцій використовуються як традиційні, так і нові засоби керування рухом курсора; перевагами такої організації взаємодії є використання звичної моторики роботи за столом і залучення образно-геометричного сприйняття людини замість абстрактно-логічного; *г) метафора редагування – єдиний стиль спілкування користувача при роботі з представленнями об’єктів* (для підтримки відповідності між об’єктами та їхніми зовнішніми представленнями пропонується стандартний набір операцій редагування: переміщення об’єкта вздовж вікна; редагування атрибутів об’єкта; видалення об’єкта; рух у середину – розкриття вмісту об’єкта; рух назовні – визначення місця об’єкта у світі об’єктів; виконання дій; вибір активного вікна; стандартні операції редагування викликаються при натисненні певних функціональних клавіш, а крім стандартних операцій, можуть виконуватися нестандартні дії, що залежать від конкретної програми); *д) інтеграція програмних засобів забезпечує комплексну інтелектуалізацію діяльності проектувальників* (необхідні засоби об’єднуються в так звані системи керування проектом чи інтегровані програмні системи й дають змогу одночасно підтримувати керування базами даних, комунікаційні можливості, засоби ділової графіки та інтелектуальний діалоговий інтерфейс; основні завдання при створенні інтегрованих систем – забезпечення інформаційного зв’язку між програмами, базами даних та екраном; перемикання активності прикладних програм; підтримка мінімального набору операцій, що надані користувачу для роботи з різними програмами; процес інтеграції функцій впливає на характер зв’язку прикладних програм та операційної системи; багато функцій операційної системи переносяться на прикладний рівень, збагачуючи і розвиваючи можливості прикладних програм [4,5]) *е) об’єктно-орієнтований підхід* (див. *об’єктно-орієнтований діалог*).

1.6. Позитивні якості інтерактивної графіки

Давньокитайське прислів’я каже: “Одна картина варта тисячі слів”. Інтерактивна комп’ютерна графіка є найважливішим технічним засобом для виготовлення та відтворення зображення з часів появи фотографії та телебачення; її додаткова перевага полягає в тому, що за допомогою комп’ютера можна отримувати зображення абстрактних, синтезованих об’єктів. Інтерактивна графіка – одна з форм взаємодії користувача з комп’ютером, в якій

об'єднані кращі риси інтерактивності текстового зв'язку з графічним взаємозв'язком. Використовуючи комп'ютерну графіку, користувач значною мірою звільняється від нудної та трудомісткої роботи, пов'язаної з виявленням загальних тенденцій шляхом перегляду багатьох текстових сторінок та цифрового матеріалу (на лістингах чи екрані).

Хоча статичні зображення в багатьох випадках є хорошим засобом для подання інформації, більші переваги забезпечують динамічно змінні зображення. А особливо це необхідно, коли потрібно зобразити в графічній формі змінні в часі явища – як реальні, так і абстрактні.

Слід відзначити, що інтерактивна комп'ютерна графіка дає змогу суттєво розширити “смугу пропускання” при спілкуванні користувача та комп'ютера за рахунок використання розумного поєднання тексту, статичних та динамічних зображень, порівняно з випадками, коли працювати можна лише з текстом. Розширення “смуги пропускання” впливає на наші можливості розуміння даних, з'ясування тенденцій та візуалізації існуючих чи уявних об'єктів. Підвищуючи ефект спілкування, графіка забезпечує зростання продуктивності праці, покращання якості й точності результатів чи зниження витрат на розрахунок та проектування.

Під час зміни чи розширення одного й більше часто завершених об'єктів на екрані та модифікації їхні візуальних характеристик інтерактивний користувач зазвичай має справу з видом цих об'єктів. Інтерактивний діалог вигляду “вивід => ввід => вивід...” між користувачем та прикладною програмою складає суть інтерактивної графіки. Така послідовна (“пінг-понгова”) модель взаємодії між користувачем та комп'ютером, у принципі, не становить труднощів для реалізації.

Робота прикладної програми полягає в моделюванні та інтерпретації даних, що вводяться користувачем.

Незалежно від того, чи будується модель в інтерактивному режимі, чи отримується як результат з іншої програми, користувач бажає мати візуальне представлення зображення одного чи декількох аспектів моделі. Прикладна програма повинна описувати для графічної системи в графічних поняттях ту частину світу, зображення якої необхідно користувачу. Незалежно від того, чи є вміст структури даних суто геометричним, вони повинні бути описані для графічної системи у вигляді вихідних графічних примітивів (крапки, лінії, ламані, ланцюжки літер), геометрично орієнтовані у двовимірному чи тривимірному просторі. Прикладна програма повинна також вказувати графічній системі, яку частину слід показати і з якої точки зору, а також на якій частині поверхні [6].

2. Реалізація діалогових форм, режимів та принципів у редакторі електричних схем системи схемотехнічного проектування “Micro-PC”

Навчально-дослідна система схемотехнічного проектування “Micro-PC” розроблялася відповідно до запропонованих принципів розробки відкритих навчально-дослідних систем [9,12], а саме: *принцип відкритості, принцип відповідності (принципи функціональної та лінгвістичної відповідності) та навчальні принципи (принцип візуалізації навчання, принцип вибору, принцип порівняльної оцінки, принцип дослідження, принцип прогляду, принцип допомоги-підказки).*

Основною умовою забезпечення при розробці системи першого принципу – *принципу відкритості* – було невикористання в системі запозичених “закритих” підсистем. Тому всі

підсистеми, що входять у систему, є власними підсистемами, які розроблено, модифіковано чи адаптовано відповідно до вимог, висунутих згідно з принципами розробки відкритих навчально-дослідних систем.

Відповідно до другого принципу – *принципу відповідності* – навчально-дослідна система схемотехнічного проектування “*Micro-PC*” характеризується як функціональною, так і лінгвістичною відповідністю.

Відповідно до *принципу функціональної відповідності*, у навчально-дослідній системі схемотехнічного проектування “*Micro-PC*” вирішуються всі типові задачі схемотехнічного проектування, властиві більшості існуючих промислових програмних систем, а саме: розрахунок статички, аналіз в частотній області, аналіз перехідних процесів, аналіз чутливості при аналізі за постійним струмом, при аналізі в частотній та часовій областях; розрахунок потужностей тепловиділення при аналізі за постійним струмом та при аналізі перехідних процесів; аналіз впливу температури середовища з врахуванням впливу як загальної температури середовища, так і локальних температурних перегрівів на тепловиділяючих елементах; аналіз, у найгіршому випадку, проведення всіх перерахованих вище аналізів у режимах заміни чи вставки елемента (параметра елемента) чи групи елементів (параметрів елементів).

Згідно з *принципом лінгвістичної відповідності* у навчально-дослідній системі схемотехнічного проектування “*Micro-PC*” описують схему за допомогою проблемно-орієнтованої вхідної мови, а керування режимами аналізу – за допомогою мови управління завданнями. І мова опису схеми, і мова управління завданнями, а також форма подання вихідних даних є традиційними і подібними, наприклад, до тих, що використовуються у системі “P-Spice”. Група директив, що реалізують навчальні принципи, дає змогу задавати команди прогляду текстів програм алгоритмів, що реалізують даний вид аналізу, підключати системні діалогові програми для відлагодження та покрокового дослідження роботи алгоритмів. Розроблені елементи вихідної мови супроводжують традиційні результати схемотехнічного аналізу інформацією про вид алгоритму, який працює на даному етапі розв’язання задачі, витрати машинних ресурсів (часу та пам’яті), точність алгоритму, його збіжність тощо. Повноцінно відтворювати запропоновані навчальні принципи допомагає як використання об’єктно-орієнтованого підходу, так і користувацький графічний інтерфейс (КГІ) системи [9].

Ми розглядаємо реалізацію графічного редактора принципів електричних схем системи “*Micro-PC*”.

2.1. Підхід до реалізації

Відправною точкою кожного зручного (чи, як кажуть, “дружнього”) інтерфейсу є метафора [4,5,7]. Адже в поняття “користувацький графічний інтерфейс” (КГІ) входить не тільки, і навіть не стільки картинка на екрані – двовимірна, тривимірна чи анімаційна або просто виконана в модному дизайні, а засоби взаємодії користувача з системою [8]. Від метафори, що закладена в КГІ, залежить ступінь розуміння і правильної інтерпретації того, що зараз відбувається на екрані. Правильно продумана й закладена метафора дає змогу не тільки легше розуміти та інтерпретувати зображення на екрані, але й позбавляє користувача необхідності кожен раз заглядати в інструкцію, щоб дізнатися, як виконується та чи інша дія, оскільки дії повинні “природно” відповідати метафорі і, нарешті, у користувача виникає почуття психологічного комфорту, характерного для зустрічі з чимось добре знайомим.

В основу метафори користувацького графічного віконного інтерфейсу навчально-дослідної системи “Micro-PC” покладено *робоче місце інженера-схемотехніка, що вивчає методи та засоби автоматизації схемотехнічного проектування* [9].

Способом підвищення ефективності схемотехнічного проектування в “Micro-PC” на етапах обробки вхідної інформації засобами інтерактивного графічного діалогового редактора електричних схем є перегляд та доробка метафори на цих етапах. Цю створену метафору можна сформулювати так: *робоче місце інженера-схемотехніка, що використовує робоче поле “макетницю” і вивчає графічні імітаційні засоби автоматизації схемотехнічного проектування*. У даному випадку в поняття “макетниця” вкладають зміст – “поверхня з рівномірною сіткою отворів, в які вставляються виводи радіоелектронних елементів, що з'єднуються провідниками”. Забезпечується надійність підключень та легкість зміни параметрів елементів та їхнього складу без жорсткого спаювання елементів. Найкраще уявити це як ескізне складання електричної схеми з реальних фізичних елементів та можливістю зовнішнього підключення вимірюваної техніки (вольтметрів, генераторів частоти, сигналів, джерел живлення, частотомірів, фазометрів тощо). Узагальнено створювана метафора має представляти лабораторію радіоаматора-початківця. Ця метафора зможе реалізуватися створенням користувацького графічного інтерфейсу для комп'ютерної системи.

У розробленні КГІ найбільшу увагу приділено формуванню інформаційного забезпечення системи, оскільки воно складає сукупність даних, які необхідні для проектування. Важливим є математичне забезпечення. Написана комп'ютерна програма реалізації вибраної метафори для підвищення ефективності використання системи схемотехнічного проектування “Micro-PC” у навчанні та дослідженнях електричних схем утворює програмне забезпечення САПР. Першим кроком практичної реалізації наведеної метафори стало створення графічного редактора принципів електричних схем, редактора бібліотечних елементів схеми, системи допомоги та функцій попереднього перегляду друку.

2.2. Використання діалогових форм, режимів та принципів

Оскільки передбачалось, що створений графічний редактор принципів електричних схем має сприяти підвищенню зручності та швидкості підготовки вхідної інформації для системи схемотехнічного проектування “Micro-PC”, то була, по можливості, закладена більша частина попередньо розглянутих принципів, типів, форм, режимів та властивостей діалогів тощо. Так, при створенні редактора бібліотек елементів схеми, вікон редагування властивостей конкретного елемента схеми, його моделі була закладена таблична форма діалогу.

Об'єктно-орієнтований діалог закладений у графічному редакторі принципів електричних схем, де кожний елемент схеми є об'єктом, має власну внутрішню структуру (умовне графічне позначення, назву, номінал, контактні виводи, назву моделі тощо). Елементи схеми вводяться, видаляються із схеми. Користувач може редагувати їхнє розташування, основні параметри, кут повороту.

Проводячи лінії зв'язків, програма вказує користувачеві, де може бути здійснене з'єднання ліній у вузол. Тоді вона працює у режимі “Ведучий – комп'ютер”. Програма автоматично нумерує компоненти схеми при їхньому введенні.

Крім узагальненої метафори графічного редактора принципів електричних схем, реалізована метафора редагування та інтеграції програмних засобів. У табл.2 наведені основні характеристики та операції редагування редактора електричних схем.

Таблиця 2

Основні характеристики редактора електричних схем *"Micro-PC" Schematic 2000*

Характеристика	Значення
Загальні характеристики	
Платформа (операційна система)	32-розрядна операційна система Windows 32 (Windows 95/98/2000, Windows NT)
Розширення, додатки	Виконувані програми (*.exe)
Мова користувацького інтерфейсу	Швидкий перехід між мовами: українська, російська, англійська
Оперування даними	Вибір курсором різного меню, контекстно-залежні засоби вибору Вибір окремих елементів складних об'єктів. Перехід на більш низький рівень ієрархії і редагування обмеженого набору параметрів
Експортування даних	Графічні формати *.bmp, *.wmf
Виведення, друк	За допомогою драйверів Windows попередній перегляд перед друком
Засоби допомоги	Поточна допомога засобами Windows, включаючи наявність гіпертексту, гіперграфіки, предметного покажчика, засоби пошуку. Файли формату *.hlp
Редактор схем	
Підтримка стандартних форматів аркушів	A-E, AO і форматів, що задаються користувачами та підтримуються встановленим друкуючим пристроєм
Одночасне відкриття декількох документів	Так
Кількість сторінок схеми	Індивідуальні файли окремих сторінок
Масштабування	Ні
Операції редагування	- Розміщення компонентів; - обертання; - монтаж (електроз'єднання); - переміщення (Drag&Drop); - видалення; - введення величини змінних; - зміна розмірів.
Візуалізація атрибутів	Індивідуальна візуалізація атрибутів та параметрів
Введення графічних рисунків	Підтримка форматів файлів *.jpg, *.jpeg, *.bmp, *.ico, *.wmf, *.emf
Введення тексту	Шрифти типу True Type
Нумерація компонентів	Автоматична при вводі елемента
Лінії	Ортогональні
Введення шин (ліній групового зв'язку)	Ні
Дискретність кута повороту компонента	90°
Складання списків з'єднань схем (Netlist) у форматах	"Micro-PC"
Редактор бібліотек схемних елементів	
Система доступу до бази даних	Borland Database Engine (BDE)
Локальна база даних	Paradox
Графічні дані	Графічні файли формату *.bmp
Введення тексту	Шрифти типу True Type
Операції редагування	- Вставка даних; - видалення даних; - зміна даних; - переміщення по записах даних.
Групування даних	Бібліотеки схемних елементів

Інтеграція з іншими прикладними програмами забезпечується через базу даних формату *Paradox* із системою доступу до баз даних *Borland Database Engine (BDE)*. Графічний редактор електричних схем може доповнюватися виконуваними програмами, команди виклику яких вбудовуються в меню та панелі кнопок. Крім того, при відповідному налаштуванні конфігурації можна викликати системні чи прикладні програми сторонніх виробників з меню редактора схем.

2.3. Реалізація

Домогтися повноцінної реалізації висунутих навчальних принципів розробки було б неможливим, якби не існувало програмних засобів об'єктно-орієнтованого проектування. Це особливо стосується принципу візуалізації навчання. Тому редактор електричних схем навчально-дослідної системи схемотехнічного проектування розроблявся з використанням засобів об'єктно-орієнтованого програмування під управлінням операційної системи *Windows 32* (тут і надалі цим терміном позначаються 32-розрядні операційні системи *Windows 95/98/NT*). Повноцінне відтворення, як вже відзначалося, запропонованих навчальних принципів забезпечується як використанням об'єктно-орієнтованого підходу, так і сукупністю діалогових форм, принципів, режимів, метафор у поєднанні з користувацьким графічним інтерфейсом (КГІ) системи цілком та окремих її підсистем. Приклад роботи інтерактивного графічного діалогового редактора електричних схем показано на рис.1.

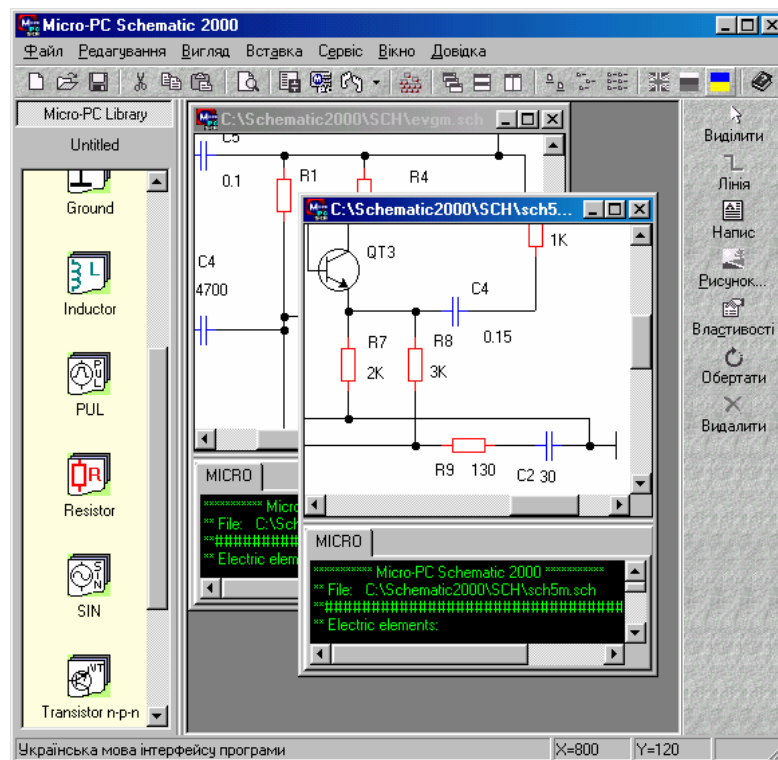


Рис.1. Приклад роботи редактора електричних схем навчально-дослідної системи схемотехнічного проектування "Micro-PC"

Редактор електричних схем для навчально-дослідної системи схемотехнічного проектування "Micro-PC" розроблявся з використанням засобів програмування *Borland Delphi 4* [10,11].

“Micro-PC” Schematic – графічний редактор принципів електричних схем має сучасні системи меню, що спливає в стилі інших програм для *Windows*, командам призначені піктограми. Для переміщення об’єкта курсором вмикають режим вибору, відмічають потрібний об’єкт і переміщують його рухом миші, поворот, видалення об’єкта виконується над виділеним елементом. Натиснення команди "Властивості" для вибраного об’єкта відкриває доступ до перегляду й редагування всіх його атрибутів та параметрів його моделі. Натиснення правою клавішею викликає контекстно-залежне меню команд. Корисна можливість зміни рухом курсора розмірів вибраного об’єкта (ліній, провідників, текстових написів та вставлених рисунків). Усі ці прийоми звичні в середовищі *Windows*, що прискорює освоєння **“Micro-PC” Schematic** і робить роботу з ним приємнішою.

Команда "Попередній перегляд" виводить документ на екран по сторінках у графічному вигляді, щоб показати, як буде виглядати документ при роздруку, що значно скорочує проміжний друк (рис.2). Передбачена можливість встановлення параметрів друку, таких, як тип принтера (тобто вибрати драйвер для друку), кількість копій при друкуванні документа, роздільну здатність, області документа, режими подачі паперу тощо.

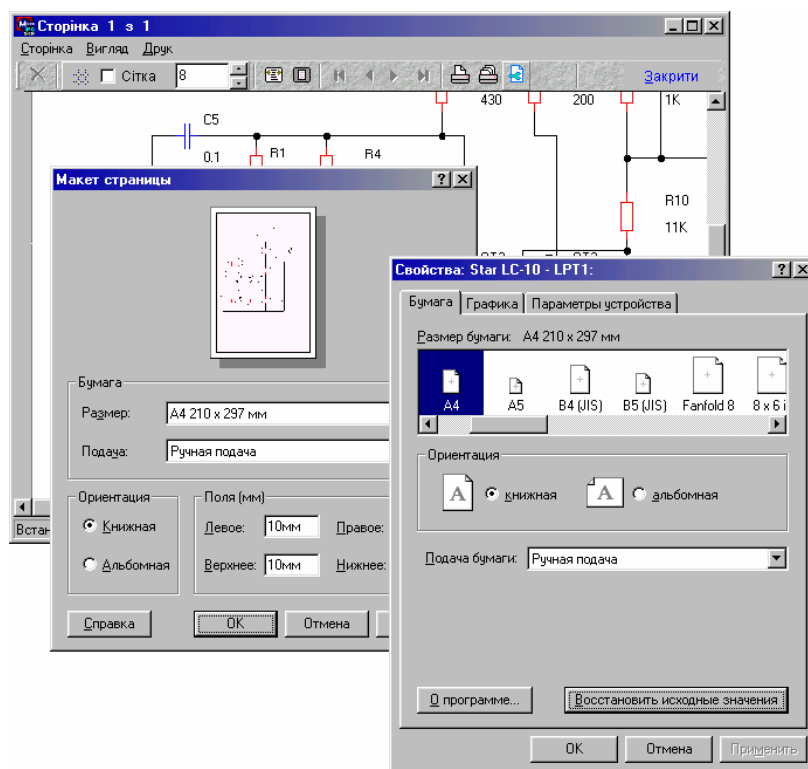


Рис.2. Вікна попереднього перегляду друку документа

“Micro-PC” Schematic надає користувачу можливість отримання оперативної допомоги, не перериваючи роботу з документом на екрані і не виходячи з **“Micro-PC” Schematic**. Таку допомогу забезпечує спеціальна допоміжна програма, яка є в *Windows 32*. Підсистема допомоги **“Micro-PC” Schematic** добре структурована й містить багато довідкових текстів. Довідка містить перелік назв розділів системи допомоги. Виконуючи фіксацію на назвах розділів, можна отримати вичерпну інформацію про можливості **“Micro-PC” Schematic** при створенні, редагуванні та друку документів. Допомога містить список усіх термінів **“Micro-PC” Schematic** за алфавітом та надає можливість пошуку необхідної інформації (рис. 3).

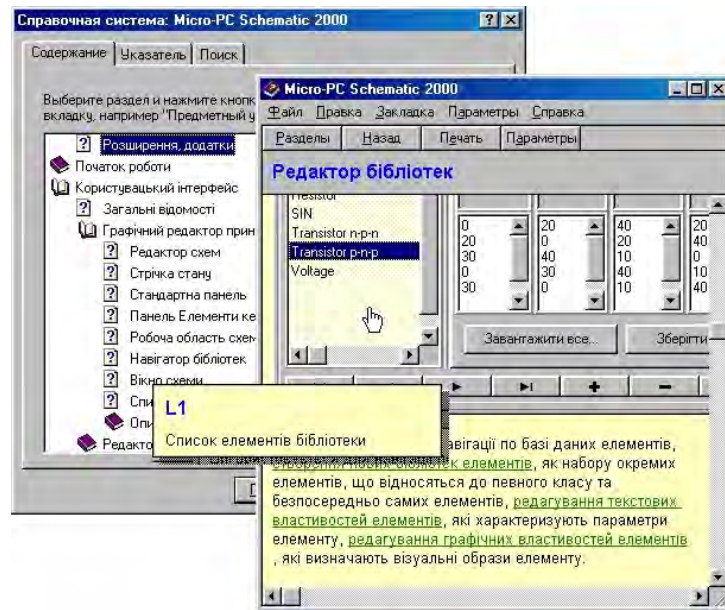


Рис. 3. Система допомоги "Micro-PC" Schematic

У будь-яке місце електричної схеми можна вставити текстові написи з різноманітним оформленням тексту шрифтами типу *True Type*, отже, є можливість застосування текстів у кирилиці.

База даних схемних елементів використовує систему доступу до даних *Borland Database Engine (BDE)*, що є набором *DLL*-бібліотек, які забезпечують низькорівневий доступ до локальних та клієнт-серверних баз даних. База даних схемних елементів виконана у форматі *Paradox*. Редактор бібліотек схемних елементів наведено на рис. 4. База даних схемних елементів для "Micro-PC" Schematic містить основні параметри, необхідні для графічно-текстового формування електричних схем та складання списків з'єднань схем.

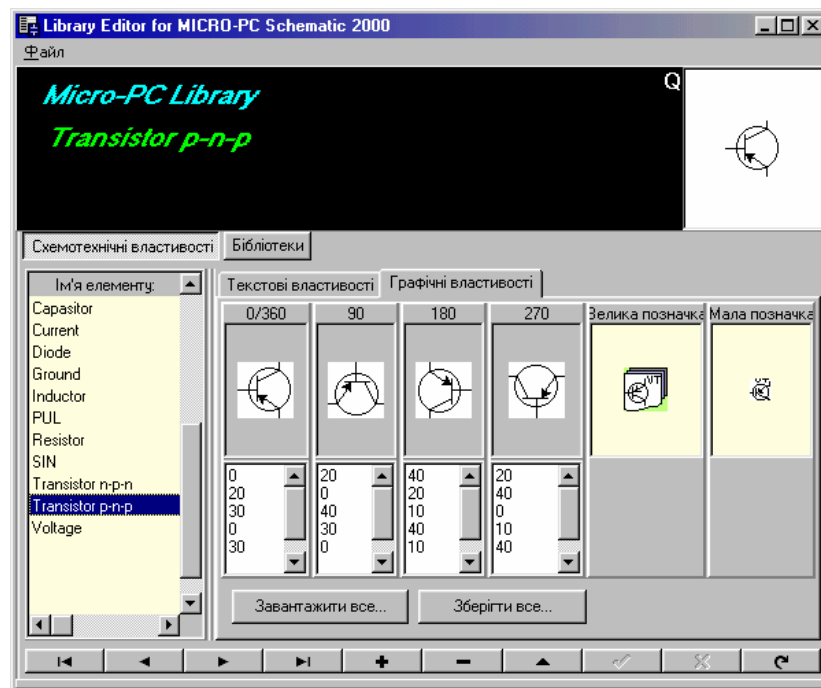


Рис. 4. Редактор бібліотечних схемних елементів "Micro-PC" Schematic 2000

3. Замість висновків: розроблення навчально-дослідної системи схемотехнічного проектування “Micro-PC” триває.

На даному етапі розроблення навчально-дослідної системи триває. Реалізації, тобто наповненню навчальними компонентами, підлягають ще ряд важливих задач етапу схемотехнічного проектування, а саме: розв’язання задачі параметричної оптимізації з обов’язковим підключенням графічного інтерфейсу підсистеми параметричної оптимізації; розв’язання задачі аналізу і синтезу допусків на параметри елементів схем; розв’язання задач забезпечення ефективності процесу оптимального проектування схем за допомогою синтезу аналітичних макромоделей з підключенням підсистеми автоматичного синтезу спрощених моделей схем на основі проведення машинних багатофакторних експериментів. Подальша розробка навчально-дослідної системи дає змогу апробувати підходи до реалізації діалогових систем, набиратися практичного досвіду студентам, адже розроблення додаткової підсистеми, як правило, є темою дипломного проекту майбутнього інженера кафедри “Системи автоматизованого проектування”.

1. Винокуров Д.И. Программирование графического диалога для автоматизированного рабочего места проектировщика АРМ-М. – Минск, 1981. 2 Мамиконов А.Г., Кульба В.В., Косяченко С.А. и др. Анализ диалоговых систем (модели и методы). – М., 1986. 3 Диалоговые системы схемотехнического проектирования / В.И.Анисимов, Г.Д.Дмитревич, К.Б.Скобелев и др.; Под ред. В.И.Анисимова. – М., 1988. 4 Разработка САПР. В 10 кн. Кн. 5. Организация диалога в САПР: Практик. пособие / В.И.Артемов, В.Ю.Строганов.; Под ред. А.В.Петрова. – М., 1990. 5 Диалоговые системы и представление знаний / Л.В.Кокорева, О.Л.Перевозчикова, Е.Л.Ющенко. – К., 1993. 6 Фоли Дж., Вэн Дэм. А. Основы интерактивной машинной графики: В 2-х кн. М., 1985. 7. Донской М. Пользовательский интерфейс. СК Пресс № 10/96. 8. Эндрю Вэн Дам. Открытые системы. Пользовательские интерфейсы нового поколения. № 6/97. С.34–37. 9. Казимира І., Близнюк М., Щербак В. Навчально-дослідна система схемотехнічного проектування “Micro-PC” як засіб навчання розробників САПР у мікроелектроніці // Вісн. ДУ “Львівська політехніка”. № 352. 1998. С.61-71. 10. Баас Р., Фервай М., Гюнтер Х. Delphi 4: полное руководство. К., 1999. 11. Фаронов В.В., Шумаков П.В. Delphi 4. Руководство разработчика баз данных. М., 1999. 12. Kazymyra I., Blyzniuk M., Lobur M. Circuit Simulation Program for Training Specialists in the Field of Circuit Design Software Development // Proc. of the 4th Int. Workshop on Mixed Design of Integrated Circuits and Systems. Poznan, Poland, 1997. P.673–679.