

<http://cgm.computergraphics.ru/content/view/44>.

3. Александров Д.Ю. 3D сканеры [електронний ресурс] / Денис Юрьевич Александров. – Режим доступу: <http://www.3dnews.ru/peripheral/3dscan>.

4. 3D Laser Scanning on a Budget [електронний ресурс] / Mike Annear, Akos Bruz, Rick Grigsby, Simeon Taylor. – Режим доступу: <http://home.exetel.com.au/mah-gar/gnu/scanner/scanneren.htm>.

5. Вежневцев В. Задача восстановления формы объекта по закраске (shape from shading) [електронний ресурс] / Владимир Вежневцев // Компьютерная графика и мультимедиа – 2004. – № 1 (5). – Режим доступу: <http://cgm.computergraphics.ru/content/view/59>.

6. С. deBoor. A Practical Guide to Splines. – New York: Springer-Verlag, 1978.

7. Крак Ю.В., Бармак О.В. Застосування нерівномірних раціональних базисних сплайнів (NURBS) в задачах синтезу та аналізу // Журнал обчислювальної та прикладної математики. – 2007. – № 1 (94). – С. 3-12.

8. Конушин А. Геометрические свойства нескольких изображений [електронний ресурс] / Антон Конушин // Компьютерная графика и мультимедиа. – 2006. – № 3 (13). – Режим доступу: <http://cgm.computergraphics.ru/content/view/141>.

9. Форсайт Дж. Машинные методы математических вычислений: Пер. с англ. Икрамова Х.Д. – М.: Мир, 1980. – 277 с.

Надійшла 19.11.2009 р.

УДК 685.3

В.П. КОНОВАЛ, М.М. ШКОДЕНКО, В.І. ЧУПРИНКА

Київський національний університет технологій та дизайну

ТИПОВІ ЗАДАЧІ, ЯКІ ВИНИКАЮТЬ ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ РОЗКРІЙНИХ СХЕМ

В роботі наведено опис типових проблем, які виникають при розробці прикладного програмного забезпечення для розв'язання задач проектування розкрійних схем шкіргалантерейної промисловості. Розглянуто етапи проектування програми, дана оцінка ефективності найбільш поширених методів програмування.

Typical problem description which arises up at development of application software for the solution of tasks of cutting out patterns of leather and foot-wear industry design is in-process resulted. The stages of program design are considered, the estimation of efficiency of the most widespread methods of programming is given.

Ключові слова: автоматизоване проектування, оптимізація.

У легкій промисловості для прискорення розв'язку оптимізаційних задач часто розробляється спеціальне програмне забезпечення. Як правило, такі додатки використовуються для автоматизації і прискорення проектувальних робіт певного виду і є вузько спеціалізованими. Значною мірою це обумовлено складністю повного охопту предметної області процесу, тому при постановці задачі важливо виділити ключові моменти і якомога детальніше встановити логічні та причинно-наслідкові зв'язки між ними. Практика показує, що тільки правильна постановка задачі, яка включає в себе побудову деталізованої моделі автоматизованого процесу, гарантує розробку ефективного програмного забезпечення, придатного для практичного використання і подальшого розвитку. При цьому власне програмування займає близько 1/5 часу і не складає принципових труднощів. В процесі розробки моделі необхідна участь спеціалістів, які мають практичний досвід у даній області. Необхідно також, щоб розробник (програміст) вільно володів термінами предметної області і мав практичне уявлення про типові виконання дій, що автоматизуються.

Для виконання поставленої завдання необхідно розв'язати наступні задачі:

- розробка технічного завдання, визначення кінцевої мети проекту;
- вибір інструментальних засобів та оцінка вартості програми і часу необхідного для реалізації;
- розробка структури програми;
- визначення ступеня інтеграції функціональних частин програми;
- обґрунтування доцільності використання об'єктного підходу;
- планування інтерфейсу, виконання контролю правильності дій користувача;
- реалізація обміну даними з іншими програмами;
- реалізація збереження даних;
- обробка вхідних даних;
- визначення форматів файлів;
- реалізація раціонального використання ресурсів комп'ютера.

Припустимо, для розробки програмного забезпечення автоматизації отримання інформації про контури шкіри і використання цих даних для формування розкрійних схем необхідно мати інформацію про наступне [1, 2]:

- види натуральних шкір, їх властивості;
- специфіка використання шкіри у різних галузях;

- обладнання і технологія розкрою;
- методи визначення пороків шкіри;
- методика побудови розкрійних схем;
- сучасні технології пов'язані з використанням і підготовкою шкір.

При побудові моделі необхідно враховувати, що неможливо завчасно передбачити всі вимоги до системи, а отже необхідно передбачити можливість її доповнення і зміни. Модель має бути гнучкою і здатною до розширення.

При розробці прикладних програм, орієнтованих, перш за все, на досягнення практичного результату слід зосередитись на засобах розробки, які не вимагають спеціальних знань і дозволяють швидко отримати готове рішення, не акцентуючись на тонкощах реалізації. Для задач різного типу існують різні засоби. Параметри для вибору в даному випадку – складність програми, вимоги до швидкодії, інтеграція з існуючим програмним забезпеченням, кількість користувачів.

Для розробки представленої програми було використано середовище програмування Turbo Delphi 2006, мова Object Pascal. Мова проста в освоєнні і використанні, придатна для написання великих програм зі складною структурою. Важливою перевагою саме цієї версії Delphi є її безкоштовність.

При написанні програмного продукту важливо наперед визначити об'єм робіт, які будуть виконуватись за її допомогою. Не великі програми (утиліти) простіші для створення, і, як правило, виконують свою функцію більш якісно. Перевагою складних програм, що об'єднують в одному інтерфейсі виконання деякої кількості різнорідних операцій є зручність у використанні – для користувача знімається необхідність перемикатися між різними додатками, які мають різний інтерфейс і контролювати обмін даними між ними. Проте створення таких програмних комплексів є на порядок складнішим, оскільки необхідно враховувати взаємодію між окремими частинами. Зростає ймовірність помилки, а доробка окремої функціональної частини часто викликає зміни у всьому програмному коді, тому для програм такого типу правильне моделювання особливо важливе.

Саме для спрощення представлення предметної області у програмному коді практично у всіх сучасних мовах програмування були реалізовані засоби об'єктно-орієнтованого програмування (ООП). ООП пропонує стандартизовані методи побудови програмного коду і гарантує, що структури створені відповідно до його вимог будуть оброблятися за правилами подібними до властивостей понять реального світу.

Головною структурною одиницею ООП є об'єкт – деяка структура коду, яка логічно відповідає поняттю предметної області програми. Наприклад для програми побудови розкрійних схем у вигляді об'єкту природно представити деталь. Відповідно до моделі (алгоритму) побудови розкрійної схеми повинні враховуватись певні *властивості* деталі – лінійні розміри, площа, координати центру, конфігурація контуру. Можна вважати, що кожна деталь виконує операції (або вони виконуються над нею, тут важливо, що дія і предмет завжди однозначно пов'язані) повороту на деякий кут, переміщення у задану точку. Тобто для поняття деталь існують певні *методи* для роботи з нею. Засоби ООП дозволяють перенести таку інтерпретацію моделі у програмний код не змінюючи її структури. Об'єкт у програмі має певні властивості, що представляють дані і методи, які виконують їх обробку. Переваги такого підходу найкраще проявляються при створенні програмних комплексів. Вдала реалізація базових понять моделі дозволяє абстрагуватися від технічних подробиць і зосередитись на реалізації алгоритмів. Оскільки об'єкт після створення дозволяє використовувати тільки наперед визначені властивості і методи, приховуючи їх реалізацію (інкапсуляція), то взагалі зникає можливість неправильно виконати дії пов'язані з поняттям (звісно за умови коректної реалізації). Один раз описавши процедуру переміщення деталі, можна викликати її у різних частинах програми, наприклад, для реалізації різних алгоритмів розміщення.

При використанні об'єктів можна враховувати відношення наслідування. Для програми побудови розкрійних схем базовим поняттям є деталь. Практично ті ж властивості (розглядаємо тільки необхідні для алгоритму) мають поняття шкіри і пороку. З точки зору програми вони представляють собою деякі геометричні фігури, що оброблюються певним чином. Можна вважати, що шкіра це деталь, яка не переміщується і може містити інші деталі, тобто шкіра має всі *властивості* деталі, а *методи* переміщення і повороту вимикаються, додається метод для визначення правильності розміщення деталі у межах шкіри. Порок можна інтерпретувати як особливу деталь, що має *властивість* – постійне розміщення на шкірі і від'ємну площу, і ті ж методи необхідні для його програмної фіксації у визначеному місці. Використовуючи поняття деталі, шкіри, пороків, можна визначити і описати структуру розкрійної схеми для натуральної шкіри.

З точки зору алгоритму розкрійна схема – це шкіра з визначеними пороками, на які не можна встановлювати деталі, і фіксований набір деталей. Для розкрійної схеми виконуються операції додавання деталі, видалення, переміщення, розрахунок кількості деталей, використаної площі, проценту ефективності розміщення. До об'єктів окрім властивостей понять предметної області необхідно задати особливості їх використання і поведінки у структурі програми. Для деталі необхідні методи відображення на екрані і на принтері. Для шкіри потрібно реалізувати розміщення пороків і врахування їх у методі визначення правильності розміщення деталі. Схема в цілому має містити засоби завантаження інформації з файлу (про деталі, шкіру, пороки), засоби для редагування своєї структури (додавання, видалення типів деталей і пороків), збереження інформації у файл, друку. Реалізація об'єктів може бути дуже складною, але надалі для виконання певної операції пов'язаної з об'єктом достатньо викликати його метод. Нижче наведено приклад запису розглянутої об'єктної моделі засобами Delphi.

```

TDetal = class // клас «Деталь»
    height, width:Integer; // висота і ширина
    center:TPoint; // центр
    kontur:array of TPoint; // інформація про межі деталі
    Function Rotate(a:Real):boolean; // поворот на кут
    Function Move(dx, dy:Real):boolean; // переміщення на задану величину
    Function ShowOnDisplay(t:TPoint; ms:Real):Boolean; // відображення
    //деталі на моніторі
end;

TPorok = class (TDetal) // клас «Порок», потомок класу «Деталь»
    // наслідує всі властивості і методи класу «Деталь»
    // обробляється при побудові розкрійної схеми особливим чином
end;

TShkira = class (TDetal) // клас «Шкіра», потомок класу «Деталь»
    // наслідує всі властивості класу «Деталь»
    // використовує власні методи і властивості
    details = array of TDetal; // деталі розміщені на шкірі
    poroku = array of TPorok; // масив пороків
    Function Is_detal_in_shkira(d:TDetal):Boolean; // визначення
    //правильності розміщення
    // деталей на шкірі
end;

TRozkriy = class // клас «Розкрійна схема»
    details = array of TDetal; //масив деталей доступних для використання
    shkira:TShkira;
    Sdet, // площа використаних деталей
    proc:real; // процент використання шкіри
    Function AddDet(d:TDetal):Boolean; // додавання деталі у схему
    Function DeleteDet(d:TDetal):Boolean; // вилучення деталі
    Function MoveDet(d:TDetal):Boolean; // переміщення деталі
    Function LoadFromFile(f:String):Boolean; // завантаження схеми з файлу
    Function SaveToFile(f:String):Boolean; // збереження схеми у файл
end;

```

Ефективне використання програми вимагає правильно побудованого інтерфейсу взаємодії з користувачем. Для прикладних додатків доцільно дотримуватись принципу концентрації уваги на основній функції програми. Якщо програма має декілька таких функцій, то необхідно одразу відображати засоби найбільш використовуваної і надавати інструмент для вибору інших засобів. У випадку побудови окремих утиліт вікно програми для побудови розкрійних схем містить панель для роботи з файлами, область для вибору деталей і основну робочу область. Для програмного комплексу базовий інтерфейс містить панель для роботи з файлами і панель вибору інструменту – відповідно до вибраного режиму роботи змінюється тип файлів. В такий комплекс можна включити розпізнавання контурів шкіри або деталі, редагування уже збереженого контуру, формування заготовки для розкрійної схеми та створення самої схеми, імпорт розкрійних схем у різні формати. Для такої системи необхідно визначити порядок використання інструментів і обмін даними між ними. У разі використання однотипних об'єктів можлива одночасна робота з різними інструментами, в протилежному – потрібно зафіксувати умови запуску кожного інструменту залежно від результатів функціонування попереднього. Скажімо не можна запустити створення розкрійної схеми поки виконується редагування деталей, що в неї входять.

При розробці багатофункціональних програм особливого значення набуває контроль правильності дій користувача. При взаємодії різних частин додатку збільшується кількість умов, які повинні бути дотримані – від правильності формату даних, що вводяться вручну, до логічної правильності результатів програми, в даному випадку розкрійної схеми. Нижче наведено деякі типові ситуації, які вимагають обов'язкового контролю:

- відкриття файлу – файл має існувати і бути правильного типу;
- ввід цифрових даних;
- контроль допустимості значення цифрових даних;
- контроль допустимих операцій – додавання типу деталі з потребою в ній рівною 0, видалення

точки з контуру, який містить тільки 3 точки і т. п.

Взагалі при визначенні допустимих дій користувача доцільно пам'ятати про правило – все, що не заборонено, дозволено, – і дозволяти тільки повністю захищені від помилок операції.

Практична цінність будь-якої програми визначається, перш за все, широтою використання її результатів. Найдосконаліша програма для побудови розкрійних схем не знайде застосування, якщо побудована схема залишиться на екрані монітора. Тому необхідно передбачити необхідні можливості для передачі даних в інші програми. Для додатку, що розглядається як приклад, виявилось необхідним реалізувати збереження схеми у власному форматі для подальшого редагування та у форматі придатному

для обробки та перегляду поширеними програмами моделювання. Зокрема у роботі [3] наведено обґрунтування збереження результуючих розкрійних схем у форматі *DXF, який є фактичним стандартом для обміну даними між САПР системами і розкрійним обладнанням і має відкриту специфікацію. Також у згаданій роботі наведено опис структури файлу. Планується також створення файлів інструкцій безпосередньо для розкрійних комплексів.

В рідких випадках вхідні дані заносяться безпосередньо в програму, як правило, існує вже попередньо підготовлена інформація оформлена у певному форматі. Від ступеня підготовленості значною мірою залежать і вимоги до самої програми. Для програми, що займається формуванням розкрійних схем, необхідна точна інформація про контур шкіри, розміщення пороків, контури деталей та їх допустиму кількість. У випадку програми з більшою функціональністю, яка була розглянута вище, достатньо мати підготовлені зображення шкіри та деталей.

Більшість задач моделювання вимагають реалізації алгоритмів, які сильно завантажують комп'ютер. Перш за все, це задачі пошуку оптимального значення деякої функції від багатьох параметрів. Для таких задач існує детально розроблений математичний апарат, де розглядаються варіанти побудови таких функцій, визначення оптимальних і достатніх умов для їх розв'язку, і власне пропонуються ефективні алгоритми. Але виявляється, що саме для типових прикладних задач, таких, наприклад, як оптимальне розміщення деталей на натуральній шкірі, з врахуванням різноманітних пороків, формального математичного розв'язку не існує. Побудована модель виявляється занадто складною задля її вирішення в рамках практичної задачі. В таких випадках використовуються більш загальні чисельні методи. Їх ефективне використання залежить, перш за все, від ефективної реалізації. При цьому необхідно враховувати загальне правило – алгоритм може працювати швидко, а може працювати економно. Економність проявляється, передусім, у використанні пам'яті комп'ютера. Основний шлях до підвищення швидкодії – винесення підготовчих операцій за межі основного циклу алгоритму, відповідно попереднє їх збереження, що вимагає додаткової робочої пам'яті. На прикладі генерації розкрійних схем по секціях для кожних двох типів деталей створювалася комбінація взаємного розміщення, яка потім використовувалася для створення секцій, з яких уже будувалася розкрійна схема. Всі три операції виконувались послідовно у вкладених циклах. Середній час генерації схем складав 15 хв. Після того, як генерація комбінацій була винесена у окрему процедуру, результати якої зберігалися у масиві (збільшилися витрати пам'яті) і передавались у готовому вигляді в головний цикл, час виконання зменшився до 7-8 хв.

Розглянувши типові проблеми, що виникають при розробці програмного забезпечення для потреб легкої промисловості, можна зробити наступний висновок – основна задача полягає у створенні максимально повної моделі, яка описує поняття і алгоритми, необхідні для створення програми. Разом з тим існують цілком достатньо розвинуті методики моделювання, що дозволяють розробити деяку проміжну функціональну схему. Така побудова виявляється зрозумілою спеціалістам у сфері, яка автоматизується, що дозволяє охопити всі аспекти технологічного процесу. Також вона досить повно охоплює логіку побудови програмного продукту, що звільняє програміста від заглиблення у подробиці технології. Практичний досвід показує, що при наявній ефективній детально розробленій моделі, час необхідний для розробки і ймовірність помилок значно зменшуються.

Важливим моментом є оптимізація програмного коду по швидкодії, оскільки вирішення практичних задач методами цифрової техніки вимагає великої кількості специфічних обчислень. Головним методом при вирішенні цієї проблеми є оптимізація самих алгоритмів.

До прикладних програм висуваються дещо специфічні вимоги щодо інтерфейсу і особливі щодо правильності результату. Такі програмні продукти мають бути максимально простими у використанні і не допускати критичних помилок у кінцевому результаті за будь-яких умов, оскільки таким результатом, як правило, є матеріальний продукт.

Література

1. Матеріалознавство виробів легкої промисловості: Підручник / Рибальченко В.В., Коновал В.П., Хом'як М.Є., Шевченко Г.І. – К.: КНУТД, 2008. – 318 с.
2. Скатерной В.А. Оптимизация раскроя материалов в легкой промышленности. – М.: Легпромбытиздат, 1989. – 144 с.
3. Чупринка В.І., Хоменко О.О., Шкоденко М.М. Підготовка вхідної інформації про схеми суміщення деталей взуття для автоматизованого розкрою // Вісник КНУТД. – 2009. – № 2. – С. 22-28.

Надійшла 13.11.2009 р.