

певного ліміту відповідно до максимальної межі довірчого інтервалу, що є резервом економії або додаткових витрат на непередбачені фактори.

Розв'язавши систему рівнянь залежностей відхилень фактичного енергоспоживання від розрахункового за числовими значеннями факторів, що стали відомими по закінченні року, можна значно підвищити точність моделі, так що похибка не перевищуватиме одного відсотка. До таких факторів можна віднести відхилення середньомісячних температур від багаторічних середніх, зміни в маршрутній системі тощо. Отже, при можливості прогнозування змін умов експлуатації точність планування потрібних обсягів електроенергії забезпечуватиметься повністю і потреба в додатковому ліміті відпадає.

Таким чином, запропонована методика визначення витрат електроенергії на пасажирські перевезення дає змогу:

- виключити планування витрат за підсумками минулого року як таке, що засновано на випадковому збігу умов експлуатації;
- визначити межі фактичних витрат електроенергії за плановими показниками роботи рухомого складу на наступний рік;
- обґрунтовано замовляти обсяги енергопостачання на наступний рік;
- отримати об'єктивні дані про економію або причини перевитрат електричної енергії за підсумками роботи в минулому році.

Отримано 28.03.2001

УДК 621.311:658.011.56

М.В.ОХРИМЕНКО

Харьковский государственный технический университет радиозлектроники

ОСОБЕННОСТИ ТЕХНОЛОГИИ РАЗРАБОТКИ ГРАФИЧЕСКИХ РЕДАКТОРОВ СХЕМ АСДУ

Рассматриваются требования к графическим редакторам АСДУ, особенности технологии их разработки. Предлагаются способы повышения быстродействия восстановления изображения экранной формы и эффективного хранения и загрузки комплекта стандартных элементов для повторного использования. Даются рекомендации для использования пакета Delphi.

В последние годы энергокомпании многих стран осуществляют реконструкцию и переоснащение диспетчерских центров (ДЦ), в том числе систем визуализации [1,2]. Одним из важных компонентов программного обеспечения автоматизированной системы диспетчерского управления (АСДУ) являются графические редакторы, обеспечиваю-

шие удобство ввода, редактирования, хранения и отображения графической информации о схемах объектов управления. Усилия разработчиков таких редакторов направлены на обеспечение возможности:

- построения связанных гипертекстовых ссылок системы на картографические планы, схемы и чертежи [3];
- построения моделей объектов, инвариантных относительно физической сущности рассматриваемых процессов [4];
- описания статических и динамических различий процессов объекта и модели [4];
- решения задач анализа текущего состояния объекта, прогноза его состояния на будущее [5];
- решения задач оперативного управления состоянием объекта.

В составе реальных проектов одновременно используется множество функций АСДУ, поэтому необходимо, чтобы каждая из подсистем (графическая, ввода-вывода, архивирования и т.п.) имела достаточное быстроедействие и ресурсоемкость.

Целью настоящей работы является решение некоторых вопросов технологии программной реализации графических редакторов для АСДУ, в частности, разработка способов повышения быстрогодействия и экономии памяти. В качестве инструмента реализации выбран пакет Delphi, являющийся средой визуального программирования с полностью интегрированной, объектно-ориентированной структурой.

Предлагаемые способы получены в результате разработки графического редактора схем RESE [6], который имеет следующие функциональные возможности: создание схем неограниченного размера с возможностью прокрутки схемы в окне; создание набора графических объектов для повторного использования; масштабирование элемента, группы элементов, схемы; поворот элемента, группы элементов; операции взаимодействия с буфером обмена; присваивание подписи (текста) элементу; изменение цветовой гаммы элемента; управление доступом; работа с подложкой (шаблоном схемы); манипулирование слоями.

В разработке графических редакторов схем можно выделить две задачи, требующие эффективного решения. Это задача восстановления графического изображения экранной формы и задача хранения и загрузки набора стандартных элементов для повторного использования.

Рассмотрим вопрос сокращения времени на восстановление графического изображения схемы электрической сети, количество объектов которой может измеряться несколькими сотнями единиц оборудования. Каждый объект представляет собой совокупность элементарных графических примитивов (линий, многоугольников, окружностей и

т.д.). Изменение параметров одного объекта, влияющих на его изображение, приводит к перерисовке всех объектов, присутствующих на схеме. Этим объясняется важность быстрodeйствия экранной формы. Положение, при котором экранная форма отображает графический образ объекта с запаздыванием, является недопустимым.

Запаздывание можно уменьшить, если система будет перерисовывать не все объекты, а только тот, параметры которого изменились. Рассмотрим реализацию предлагаемого способа в среде Delphi. Все графические операции в Delphi выполняются с использованием свойства Canvas, которым обладают классы TForm и TPaintBox. Класс TForm является наиболее общим и известным из всех объектов библиотеки визуальных компонентов и представляет собой форму, изображенную на рис.1,а. Класс TPaintBox, показанный на рис.1,а, является элементом управления окном рисования и представляет собой прямоугольную область на форме, где могут проводиться операции рисования. Событие OnPaint классов TForm и TPaintBox вызывается, когда содержимое объекта требует преобразования и, как следствие, перерисовки.

Следовательно, если поместить операторы рисования графических примитивов в событие OnPaint класса TForm, размер области перерисовки будет соответствовать размеру формы, как это показано на рис.1,б. В случае нахождения операторов в OnPaint класса TPaintBox размер области равен размеру прямоугольной области вокруг элемента (рис.2,б).

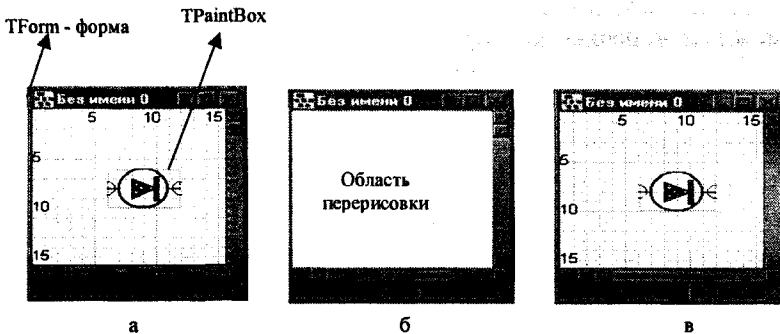


Рис.1 – Последовательность восстановления изображения всех объектов класса TForm

Сопоставление площадей областей перерисовки на рис.1,б и 2,б показывает, что предлагаемый способ использования объекта класса TPaintBox способствует уменьшению времени на восстановление изображения и повышает быстрodeйствие системы в целом.

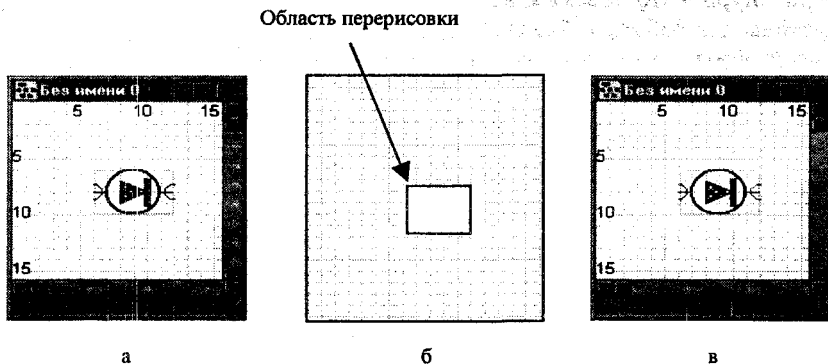


Рис.2 – Последовательность восстановления изображения класса TPaintBox

Рассмотрим задачу загрузки и хранения комплекта инструментальных средств, который включает набор графических объектов для повторного использования. Каждый раз при запуске редактора выполняется операция загрузки комплекта на панель инструментов, один из вариантов которой изображен на рис.3. Операция загрузки состоит в создании графического образа для всех объектов, хранимых в базе данных стандартных элементов. Существенными являются скорость выполнения этой операции и объем памяти, которую занимает комплект инструментальных средств.

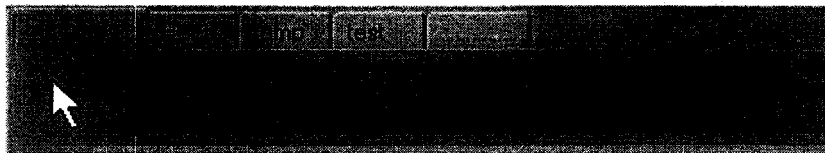


Рис.3 – Пример панели инструментов

Способ хранения и загрузки объектов в графическом формате bitmap является неэффективным по соображениям быстродействия и ресурсоемкости. Предлагается создать текстовый формат данных, представляющий собой совокупность примитивов объекта, записанных в определенном порядке. Примитивы – это графические функции рисования линии, окружности, многоугольника, дуги, сектора совместно с их параметрами (например, толщина линии, цвет и т.д.). Рассмотрим реализацию предлагаемого способа в среде Delphi. В качестве платформы для создания панели инструментов используется компонент Panel (Панель). Panel содержит объекты – кнопки SpeedButton, графическое изображение на которые помещается с помощью свойства

Glyph. Glyph – это объект класса TBitmap, инкапсулирующий способы и методы для работы с битовой картой (bitmap). Таким образом, визуальный образ элемента можно получить двумя способами с помощью следующих методов TBitmap: методом LoadFromFile, который загружает графический файл с диска и методами рисования графических примитивов свойства Canvas, последовательный вызов которых формирует изображение.

Эти два способа получения визуального изображения были избраны для тестирования, которое проводилось на компьютере следующей конфигурации: CPU Celeron-500; 128 RAM. Целью тестирования было сравнение основных характеристик указанных способов хранения и загрузки элементов как совокупности примитивов. Результаты тестирования приведены в таблице.

	Количество стандартных элементов	Время создания графического образа	Объем памяти для одного элемента
Метод LoadFromFile	100000	2 мин 33 с	3238 байт
Методы свойства Canvas	100000	1 мин 12 с	285 байт

Анализ экспериментальных данных показал, что предлагаемый метод с использованием свойства Canvas превосходит по быстродействию метод LoadFromFile приблизительно в два раза и требует для хранения одного элемента в десять раз меньше памяти ЭВМ.

Таким образом, предлагаемые методы позволяют повысить быстродействие восстановления экранной формы и освободить часть ресурсов памяти ЭВМ для других подсистем АСДУ.

1. New control center for Ukraine // Int Journal Ydropower and Dams. – 1999. – №3. – С.12.
2. Roche Peter, Connolly Eamonn. Challenges and strategies for refurbishment of Power system control centers/ *Energiya*. – 1999. – 48, № 1. – С.3-11.
3. Варлатая С.К., Зенькович Д.А. Автоматизированная система визуализации электрических схем электроснабжения // Тр. Дальневост. гос. техн. ун-та. – 1999. – №122. – С.10-11.
4. Архипова Е.Н. Требования к математическим моделям энергетических объектов, используемым в современных тренажерах для оперативного персонала // Энергосбережение и водоподготовка. – 1999. – №4. – С.72-78.
5. Shin J.-R., Lee W.-H., Im D.-H. A windows-based interactive and graphic package for the education and training of power system analysis and operation // IEEE Trans. Power Systems. – 1999. – 14. № 4. – Р. 1193-1199.
6. Дударь З.В., Охрименко М.В., Диалло Т.А. Редакторы схем в составе прикладного программного обеспечения АСДУ // Вестник ХПИ. Вып.125. – Харьков: ХПИ, 2000.

Получено 10.04.2001