

На графике ниже показаны пространственные колебания типа Фриделя, присутствующие как в профилях электронной плотности, так и в профилях потенциальной энергии.

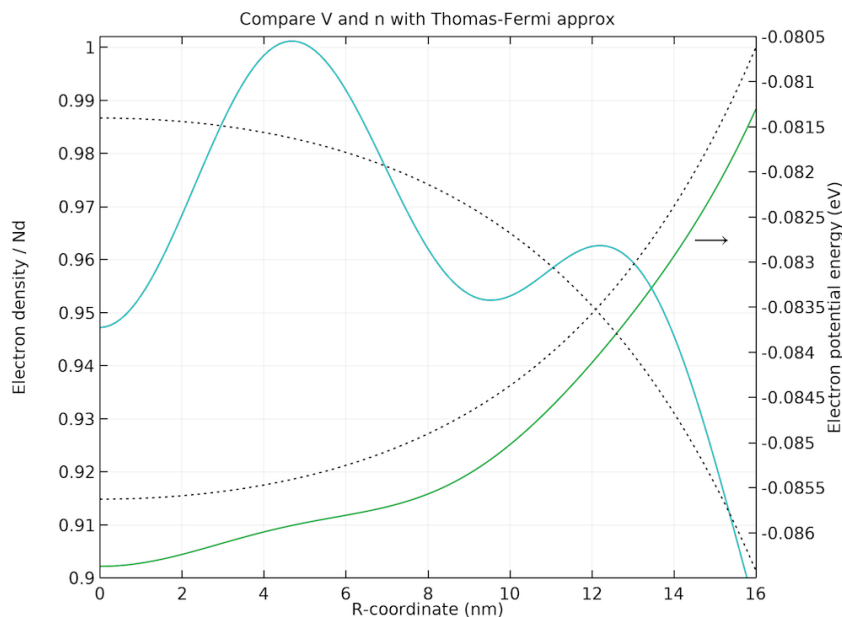


Рисунок 2 – Увеличенный график пространственных колебаний типа Фриделя в профилях электронной плотности и потенциальной энергии.

В этом исследовании мы продемонстрировали, что интерфейс уравнения Шредингера-Пуассона и тип исследования Шредингера-Пуассона упрощают настройку и решение системы Шредингера-Пуассона с использованием самосогласованных результатов Шредингера-Пуассона для модели нанопроволоки GaAs.

#### Список литературы

1. Luscombe, J.H. Electron confinement in quantum nanostructures: Self-consistent Poisson-Schrödinger theory / J.H. Luscombe, A.M. Bouchard, and M. Luban // Phys. Rev. B. – 1992. – vol. 46, No. 16. – P. 10262.

УДК 681.3

### О ПОСТРОЕНИИ КАРКАСОВ ПРОГРАММ С УЧЕТОМ ПРЕДЪЯВЛЯЕМЫХ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

*В. В. Шпак*

*Брестский государственный технический университет, Брест  
Научный руководитель: Г. Л. Муравьёв, к. т. н., доцент*

При проектировании и разработке программ, приложений исходят из имеющихся требований, предъявляемых к характеристикам продукта. Это функциональные требования, задающие закон функционирования - набор функций, реализуемых программой и определяющих связь входных данных с выходными, требования к характеристикам исполнения, для оконных программ - требования к организации и дизайну интерфейсов (ГИП) и др. [1].

В настоящее время отмечены оформившиеся тенденции проектирования приложений в широких сферах применения, таких как экономика, социальные учреждения и т. п. [1, 2]: стремление к повышению продуктивности разработки с соблюдением необходимой степени корректности проектных решений даже в ущерб показателям эффективности выполнения программ, которая при необходимости достигается последующим рефакторингом.

Указанное предполагает использование таких элементов технологий активной разработки [3]:

- как раннее вовлечение в процессы проектирования заказчиков, использование их экспертных знаний, в том числе для анализа задач, функций, тестирования решений;

- опережающее моделирование продукта, создание действующих макетов разной степени функциональной полноты, в том числе с учетом элементов технологии структурной разработки [4], применяющей нисходящее проектирование и использующей для получения макетов подпрограммы, с разной степенью детализации отображающих исполняемые функции (от подпрограмм-заглушек до готовых функций).

Здесь макет как модель программы представляет собой исполняемый шаблон – функциональный каркас приложения (ФКП), построенный на базе выбранного типового системного автокаркаса приложения (КП) и специализированный с учетом требований к продукту, языку программирования, платформе, библиотекам.

Также следует отметить:

- рост составляющей рутинных операций в разработке программ, в том числе в части работы с оконными интерфейсами (окнами, иерархиями окон, элементами управления, обработчиками и т.п.);

- значительную степень автоматизации средств разработки, систем программирования, обеспечивающих использование каркасного программирования, когда разработчик получает автоматически сгенерированный текст программы – каркас приложения, реализующей системные функции, рутинные операции; поддержку визуального проектирования и реализации интерфейсов.

Все это создает предпосылки для автоматизации и быстрого построения действующих макетов программ.

В работе предметная область – этап объектно-ориентированного анализа жизненного цикла разработки программ в части автоматизации использования объектных моделей. Цель работы – макетирование приложений на базе результатов анализа предметной области, построение ФКП, анализ успешности принятых решений, выработка рекомендаций по созданию макетов.

Используемый теоретический аппарат: структурный подход, процедурная и объектная парадигмы для описания каркасов, порождающая парадигма для их генерации.

Представленные результаты получены средствами Microsoft Visual Studio. Соответственно объект рассмотрения - типовые каркасы (WinAPI, MFC, Windows Forms) в части их использования для автоматической донастройки и специализации до уровня ФКП с учетом функциональных требований.

Требования могут быть специфицированы с разной степенью детальности - от неформализованного, текстового описания, до описаний в терминах объект-

ной парадигмы с представлением ожидаемых действий (на базе классов, объектов и привязанных к ним атрибутов, свойств, методов и т.д.).

Указанное позволяет генерировать и добавлять к КП: меню, кнопочное меню; окна, иерархии окон, элементы управления в составе интерактивных окон; прототипы и методы-обработчики оконных сообщений; прототипы и методы-обработчики для бизнес-логики приложения; шаблоны классов и иерархий пользовательских классов.

Сформулированы проблемы и способы формализации описаний потоков событий функций программ. Результаты могут использоваться разработчиками ПО для быстрого моделирования проектов, предоставления макетов заказчику; в учебном процессе для повышения эффективности обучения проектированию приложений, анализу предметных областей.

#### **Список литературы**

1. Орлов, С. А. Программная инженерия / С. А. Орлов. — СПб.: Питер, 2016. — 640 с.
2. Эванс, Э. Предметно-ориентированное проектирование: структуризация сложных программных систем / Э. Эванс. - М.: ООО “И.Д. Вильямс”, 2016. - 448 с.
3. Активное программирование [Электронный ресурс]. – 2021. – Режим доступа: [https://ru.wikipedia.org/wiki/ Экстремальное программирование](https://ru.wikipedia.org/wiki/Экстремальное_программирование). – Дата доступа: 10.10.2021.
4. Хьюз, Дж. Структурный подход к программированию / Дж. Хьюз, Дж. Мичтом. М.: Мир, 1980. - 278 с.