

# Автоматизированная система DOERIS. Расчёт элементов микро-оптики

Л.В. Яблокова  
Самарский национальный  
исследовательский университет им.  
академика С.П. Королева  
Самара, Россия  
ylv@ssau.ru

И.В. Ветлова  
ИнтеллектСофт  
Самара, Россия  
irinka2006.82@mail.ru

А.Г. Дмитриев  
ИнтеллектСофт  
Самара, Россия  
admitriyev@inbox.ru

В.В. Прокофьев  
ИнтеллектСофт  
Самара, Россия  
prokofievvladimir@gmail.com

Д.Е. Яблоков  
ИнтеллектСофт  
Самара, Россия  
dyablokov.SSAU@gmail.ru

**Аннотация** — В статье рассматриваются базовые принципы конструирования программного обеспечения, применённые в рамках проекта по проведению модельных экспериментов при расчёте элементов микро-оптики. С точки зрения классического подхода к организации архитектуры, послойно рассматривается каждый уровень программного решения с описанием его функциональных особенностей.

**Ключевые слова** — автоматизированная система, DOERIS, синтез элементов микро-оптики, расслоение системы, конструирование программного обеспечения

## 1. ВВЕДЕНИЕ

Без автоматизированных систем обработки и представления данных трудно представить современные научные исследования, образование, передовые производственные технологии. Немаловажную роль, при этом, играет наличие возможности проведения модельных экспериментов, позволяющих сформировать критерии оценки ожидаемого качества для исследуемых объектов.

Автоматизированная система DOERIS (Diffractive Optical Elements Research Information System) предоставляет такую возможность для предварительной оценки качества будущих оптических элементов с точки зрения используемых при расчёте варьируемых технологических параметров и параметров специфичных для конкретного элемента.

В статье рассматривается первый этап проекта автоматизированной системы DOERIS, направленный на расчёт и визуализацию результатов для нескольких типов оптических элементов [1]: дифракционная сферическая линза, дифракционная цилиндрическая линза, радиально-симметричный аксикон, винтовой аксикон, бинарная дифракционная решётка и дифракционная призма. Далее будут представлены некоторые категории идей и базовых принципов, положенных в основу конструирования проекта DOERIS.

## 2. АРХИТЕКТУРА

В виду возможного наращивания уровня сложности [2] в процессе развития функционала DOERIS, а также необходимости при сопровождении в адаптации к вновь поступающим требованиям было принято решение использовать усовершенствованный вариант (рис. 1) классической архитектурной идиомы по расслоению системы [3].

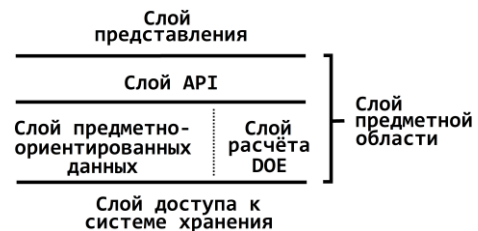


Рис. 1. Расслоение системы

Оправданность такого подхода заключается в том, что каждый слой отвечает только за небольшой, узкоспециализированный сегмент, а полученную многоуровневую архитектуру намного легче интерпретировать и поддерживать.

## 3. СЛОЙ ДОСТУПА К ДАННЫМ

На данном уровне реализованы интерфейсы доступа к данным, обеспечивающие работу всей системы. Основной задачей слоя является предоставление информации о состояниях, характеристиках, измерениях и иных свойствах, которыми обладают объекты системы, используемые на других слоях. Можно выделить несколько информационных фрагментов, которые лежат в области ответственности слоя доступа к данным. Это: работа с пользователями и ролями; работа с проектами, рабочими пространствами и результатами расчёта; работа с данными по оптическим элементам и их параметрами. В основном для каждого из информационных фрагментов реализованы CRUD методы [4] с небольшими дополнениями по способу извлечения данных.

## 4. СЛОЙ ПРЕДМЕТНОЙ ОБЛАСТИ

Слой предметной области сочетает в себе несколько взаимосвязанных слоёв для улучшения качества иерархического взаимодействия между уровнями.

Слой API отвечает за обработку запросов от внешней среды в том числе и слоя представления с целью их ретрансляции в более низкоуровневые конструкции. По сути слой API является фасадом [5] изолирующим клиентскую сторону от технических деталей более специализированных программных компонентов.

Слой расчёта отвечает за вычисление фазовой функции для конкретного оптического элемента. Функционал расчёта реализован с использованием наиболее распространённого типового решения для настраиваемого поведения алгоритма. Шаблонный

метод [5] задаёт строгую основу, позволяя подклассам переопределять некоторые шаги алгоритма, не затрагивая его структуру в целом (рис. 2).

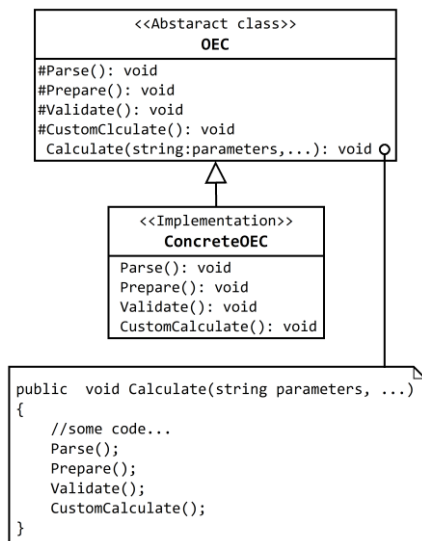


Рис. 2. Диаграмма типового решения для расчёта фазовой функции

Слой предметно-ориентированных данных содержит спецификации объектов системы, которые отвечают за логику сохранения или извлечения информации из базы данных, инкапсулируя излишние детали низкоуровневой реализации, способ переноса данных [3] (сериализация и десериализация) между взаимодействующими интерфейсами, принадлежащими различным слоям. В большинстве случаев основной задачей присутствующих на этом уровне классов является описательный аспект предметной области (рис. 3).

XmlDSL	XmlDRSA
+WaveLength: string	+ApexHalfAngle: string
+FocalDistance: string	+ApertureRadius: string
+ApertureRadius: string	+Resolution: XmlSize
+Resolution: XmlSize	+WorkdapceId: string
+WorkdapceId: string	+UserUnits: string
+UserUnits: string	

Рис. 3. Примеры xml-преобразования данных для расчёта DOE

Как видно из рис. 3, объекты xml-преобразования данных для расчёта различных элементов обладают почти идентичной сигнатурой. Они создаются как элементы модели предметной области, о которых достаточно знать узконаправленным компонентам, ответственным за выполнение конкретных операций. Также на уровне предметно-ориентированных данных на данной стадии проекта присутствуют подобные реализации для преобразования JSON.

## 5. СЛОЙ ПРЕДСТАВЛЕНИЯ

Этот уровень охватывает всё, что имеет отношение к общению пользователя с системой (рис. 4). Он служит для отображения состояний объектов и интерпретации подаваемых пользователем команд в операции API.

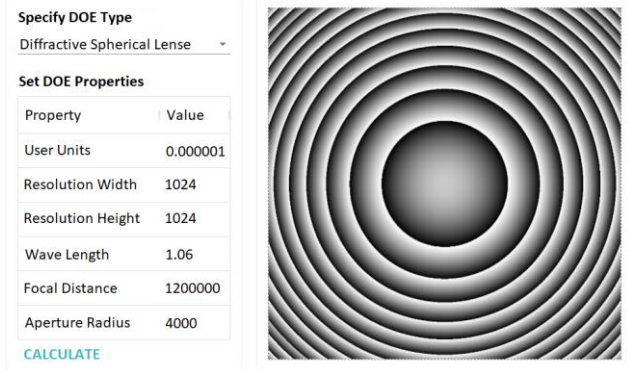


Рис. 4. Окно UI по расчёту дифракционной сферической линзы

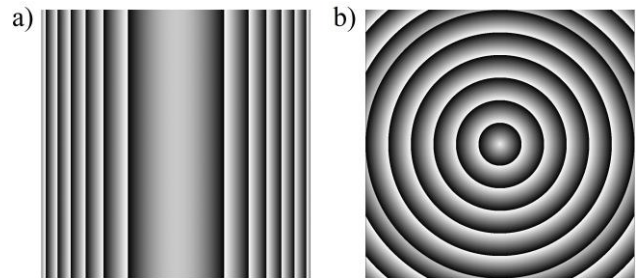


Рис. 5. Результаты модельных экспериментов для некоторых поддерживаемых элементов, где а – дифракционная сферическая линза, б – радиально-симметричный аксикон

Фактически здесь реализуется стратегия по выбору способа обработки, подразумевающая возможность изменения параметров расчёта для поддерживаемых элементов (рис. 5), а также комплексное исполнение с вовлечением нескольких уровней, таких как, например, сама операция расчёта (рис. 2).

## 6. ЗАКЛЮЧЕНИЕ

На данном этапе проекта были реализованы расчёт и визуализация результатов для некоторых типов элементов. В дальнейшем планируется расширение функционала DOERIS как с точки зрения увеличения количества типов поддерживаемых элементов, так и с позиций развития по работе с результатами вычислений (получение срезов, преобразование в технологические форматы и т.п.)

## БЛАГОДАРНОСТИ

Особую благодарность выражаем д.ф.м.н., профессору Павельеву В.С. за своевременные корректировки и рекомендации в ходе работы над проектом.

## ЛИТЕРАТУРА

- [1] Сойфер, В.А. Дифракционная компьютерная оптика / В.А. Сойфера – М.: «Физматлит», 2007. – 736 с.
- [2] Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч – М.: Вильямс, 2008. – 718 с.
- [3] Фаулер, М. Архитектура корпоративных программных приложений.: Пер. с англ. – М.: «Вильямс», 2006. – 544 с.
- [4] Speelpenning, J. Data Modeling and Relational Database Design: Student Guide / J. Speelpenning, P. Daux, J. Gallus – Redwood City (California): Oracle Corporation, 2001. – 320 p.
- [5] Гамма, Э. Приёмы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма и др. – М.: Питер, 2014. – 366 с.