УДК 004.415.28

Т.М. Буланая, В.В. Подлесняк

МОДЕЛИРОВАНИЕ ГИБКОГО ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ОБРАБОТКИ ДАННЫХ

Актуальность. Основным недостатком существующих программных продуктов обработки данных является то, что при работе возникает необходимость написания программ языком пакета, который не очень удобно для медика или других специалистов, не имеющих навыков программирования и слабо ориентирующихся в статистических методах.

Анализ известных исследований. Для обработки данных обычно используют такие мощные платформы и пакеты как SAS [1], SPSS [2], STATISTICA [3], Maple [4], MathCAD [5], Microsoft Excel и NeuroPro [6]. Основным достоинством программных комплексов является наиболее широкий охват статистических методов, регрессионного, факторного, кластерного анализа и не только, что удачно совмещается с большим количеством удобных средств визуализации результатов обработки, притом не все пакеты имеют возможность проведения нейросетевого моделирования. Также, не смотря на все свои позитивные качества, данные программные продукты имеют большую цену, являются слишком громоздкими, не приспособлены под конкретные области применения и не доступны широким массам. Поэтому исследователям не всегда легко интерпретировать полученный результат.

Постановка задачи. Таким образом, имеется острая необходимость в гибком программном комплексе для обработки данных, который будет:

– простым для понимания экспертов в различных науках;

– имееть удобный и дружественный интерфейс, позволяющий пользователю минимизировать время на обучение работе;

– адаптироваться под конкретную предметную область;

– иметь возможность быстрой реконфигурации при изменении спектра решаемых задач;

 ISSN 1562-9945

– проводить подсчеты стандартных и специфических для предметной области пользователя показателей;

– избавлять пользователя от необходимости разбираться в нюансах работы системы;

– предоставлять визуализацию последовательности обработки данных;

– конфигурироваться на этапе инсталляции;

– иметь возможность расширения функциональности при помощи встраивания новых частей без переустановки и перекомпиляции кода.

Основная часть. Для реализации поставленной задачи был смоделирован гибкий программный комплекс MiningLibs.

Для обеспечения успешности разрабатываемого продукта, необходимо грамотно подобрать архитектуру. На данный момент существуют следующие архитектурные шаблоны [7]: объектно-ориентированная архитектура (Object-Oriented Architecture), архитектура модель-вид-контроллер (Model-View-Controller Architecture), канальная архитектура (Pipes and Filters Architecture), многоуровневая архитектура (Layered Architecture). Наиболее распространенным подходом к разработке архитектуры является рассмотрение архитектуры на нескольких уровнях абстракции по своему собственному шаблону, которые характеризуются постепенной детализацией системы, такая архитектура носит название гетерогенной. В спроектированной системе была использована комбинация архитектурных подходов (рис. 1).

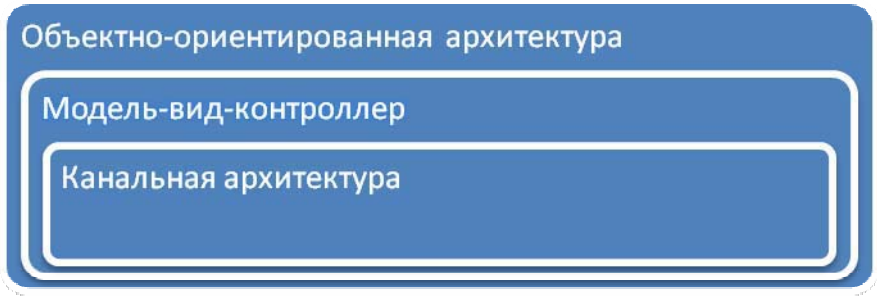

Рисунок 1 - Архитектурная схема программного комплекса MiningLibs

Для обеспечения независимости компонент системы была применена объектно-ориентированная архитектура. При данном

подходе компоненты системы рассматриваются как объекты. Объект представляет собой некоторую сущность, что содержит в себе данные и операции, которые могут быть выполнены над этими данными. Кроме интеграции данных и операций, для объектов характерна защита своих данных от непосредственного доступа со стороны других объектов.

Для разделения функциональности между частями системы была применена архитектура MVC (Model-View-Controller) (рис. 2).

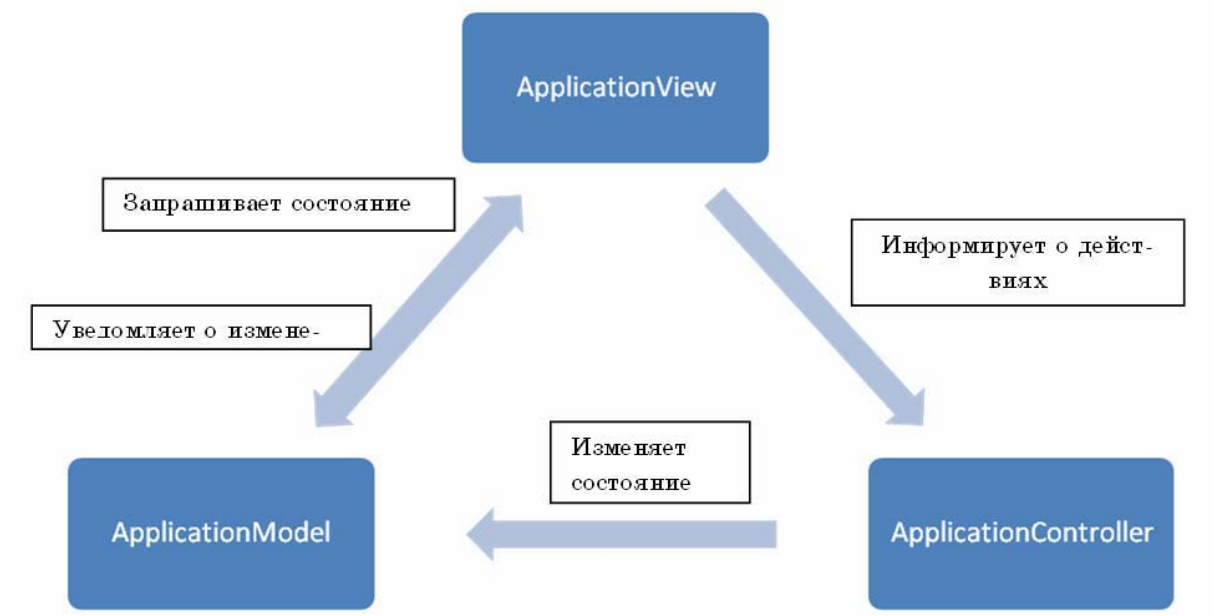

Рисунок 2 - Схема ядра программного комплекса MiningLibs

Основными архитектурными компонентами ядра системы являются модель (ApplicationModel), вид (ApplicationView) и контроллер (ApplicationController). ApplicationModel хранит в себе модули, предназначенные для обработки данных, а также взаимосвязи между этими модулями. ApplicationView является визуальным представлением модели, которое подается пользователю через графический интерфейс. ApplicationController определяет поведение всей системы и реализует алгоритм ее функционирования. Для построения визуального представления данных ApplicationView запрашивает у ApplicationModel ее состояние и формирует соответствующее изображение. О каждом действии пользователя ApplicationView информирует ApplicationController, который на основании заложенного алгоритма функционирования вносит соответствующие изменения в ApplicationModel. ApplicationModel

 ISSN 1562-9945

информирует ApplicationView о изменении своего состояния, ApplicationView обращается к ApplicationModel с запросом о новом состоянии и корректирует соответствующим образом свое визуальное представление.

Для обеспечения гибкости системы было принято решение представить модули с их взаимосвязями как канальную архитектуру. В данном подходе ApplicationModel рассматривается как контейнер модулей (Unit) и их взаимосвязей. Каждый модуль выполняет строго специализированные задачи и имеет множество входов и множество выходов, причем форматы данных на входах и выходах специфицированы, но при этом не может определять, с какими именно другими модулями может быть соединен. В процессе функционирования модуль считывает данные со своих входов, обрабатывает считанные данные, соответственно к заложенной функциональности и выдает обработанные данные на выход.

В данном подходе можно выделить несколько основных модулей: Unit загрузки данных, Unit предварительной обработки данных, Unit моделирования, Unit вывода результатов, которые взаимодействуют по канальной схеме (рис. 3).

Рисунок 3 - Схема потока данных программного комплекса MiningLibs

Построение и конфигурацию цепочки модулей производит пользователь. Модули представлены пользователю в виде графических компонент, имеющих средства настройки и помощи. Добавление и связывание компонент проводится с применением технологии Drag&Drop.

Для разработки системы выбран язык программирования Java. Основным преимуществом Java является кроссплатформенность, также язык поддерживает объектно-ориентированную парадигму программирования. В качестве среды разработки применяется Eclipse

Platform. В процессе разработки были использованы пакеты Spring framework, предоставляющие расширения к стандартным средствам разработки, и JFreeChart для вывода графиков. На этапе разработки системы тестирование проводится при помощи утилиты JUnit. Интеграционное тестирование проводится конечными пользователями.

Разработанная система позволяет проводить интеллектуальный анализ данных (Data Mining), таким образом, помогая пользователю выявлять полезные скрытые взаимосвязи в больших объемах данных. Для этого применяется как машинное обучение, так и расширенные средства визуализации.

Для обеспечения гибкости формат входных данных не фиксирован и зависит только от подключенных к системе модулей загрузки, которые, при необходимости, могут быть обновлены и расширены. Таким образом, для загрузки будут применяться как текстовые файлы, так и базы данных. Формат выходных данных также не специфицирован и зависит только от модулей вывода результатов. Данный подход позволяет сохранять результаты работы не только в виде файлов различных форматов и баз данных, но также в виде графиков и схем.

Система поставляется конечному пользователю в виде дистрибутива, требующего инсталляции. На этапе инсталляции система производит оценку задач, которые будет решать пользователь и на основании этого конфигурирует доступные пользователю модули. В дальнейшем система при запуске проводит самодиагностику и информирует пользователя о неполадках, если таковые имеются. Изменение функциональности система производит по запросу пользователя. Имеется возможность проведения поиска обновлений.

Пример построения модели в программном комплексе MiningLibs: оценить применимость использования MultiLayer Perceptron (MP) многослойной нейронной сети при гипертонической болезни [8]. Данные предоставляются в виде текстового CSV файла. Для решения данной задачи используется для моделирования модуль MP, оценка адекватности проводится при помощи модуля ROC-анализа. Для загрузки данных используется загрузчик CSV файлов. Также используется модуль нормализации данных для обеспечения

 ISSN 1562-9945

работы аппарата МР. Таким образом, имеем следующую схему потока данных (рис.4):

Рисунок 4 - Схема потока данных в MiningLibs решении задачи диагностики

Выводы. Был смоделирован гибкий программный комплекс обработки данных, который имеет возможность наращивания функциональности без перекомпиляции и переустановки, а также адаптируется под конкретную предметную область конечного пользователя и предоставляет удобную визуализацию, как процесса моделирования, так и полученных результатов. Программный комплекс MiningLibs позволяет проводить анализ данных статистическими методами, методами корреляционного, регрессионного, факторного, кластерного анализа, производить подсчеты стандартных статистических показателей. Данный пакет имеет возможность проведения нейросетевого моделирования, подсчет нестандартных коэффициентов и показателей, а также позволяет обрабатывать данные различных форматов. Таким образом, система сочетает в себе позитивные качества систем данного класса, а также привносит новый взгляд на сам процесс проведения анализа данных и предоставления пользователю результатов работы. Программный пакет был протестирован на данных исследований газочувствительных свойств оксиднометаллических полупроводников к парам спиртов, а так же в рамках Национальной Программы профилактики и лечение артериальной гипертензии [8].

ЛИТЕРАТУРА

1. http://www.sas.com/offices/europe/russia/software/
2. http://www.spss.com.ua/
3. http://www.statsoft.ru/home/portal/
4. http://www.maplesoft.com/products/Maple/
5. http://www.ptc.com/products/mathcad/data-analysis-extension-pack
6. http://www.neuropro.ru/

7. Buschmann F. Pattern – Oriented Software Architecture/ Buschmann F., Meunier R., Rohnert H., Sommerlad P., Stal M. // A System Of Patterns Vol.1, Wiley: - New York, 1996. – P. 457.
8. Буланая Т.М. Автоматизированная система комплексного индивидуального анализа при диагностике сердечно-сосудистых заболеваний. / Буланая Т.М, Колесник Т.В. // Математичне та програмне забезпечення інтелектуальних систем MPZIS-2006: Міжн. наук.-практ. конф., 15-17 лист. 2006 р.: тези доп. – Дніпропетровськ, 2006. – С. 81.

Получено 16.11.2008г.

 ISSN 1562-9945