

УДАЛЕННАЯ СИСТЕМА АНАЛИЗА ДАННЫХ ФЛУОРЕСЦЕНТНОЙ СПЕКТРОСКОПИИ

Ю. П. Жишко, А. В. Дигрис

Белорусский государственный университет, Минск, Беларусь

Методы флуоресцентной спектроскопии находят в современной науке широкое применение для исследования свойств сложных молекулярных систем [1]. Для получения информации об исследуемом образце во флуоресцентной спектроскопии существует ряд экспериментальных методов: метод счета единичных фотонов, фазово-модуляционный метод, метод флуоресцентной флуктуационной спектроскопии. Перечисленные методы позволяют получать ряд экспериментальных кривых, которые несут информацию об исследуемом образце. Последующий анализ экспериментальных данных позволяет получать различные характеристики исследуемой системы, такие как частота столкновений с молекулами тушителей, скорость переноса энергии, скорости реакций в возбужденных состояниях, коэффициент диффузии молекул т. д. [1].

Поскольку экспериментальные данные содержат статистический шум, для проведения их анализа требуется использование набора специально разработанных для этой цели алгоритмов, каждый из которых выполняет достаточно четко определенную отдельную задачу. Процесс анализа включает подготовку исходных, построение модели, генерацию начальных приближений, подгонку теоретической кривой к исходной с использованием различных целевых критериев и методов оптимизации [2].

Задача создания программного обеспечения, предназначенного для выполнения обработки экспериментальных данных, требует не только хорошего знания математических алгоритмов анализа данных, но и умения создавать довольно сложные программные комплексы, способные выполнять не только анализ данных, но и обеспечивать хранение, как измеренной информации, так и результатов проведенного анализа. Решение такой задачи в пределах каждой научной лаборатории, занимающейся прикладными исследованиями, часто становится нерациональной. По приведенным выше причинам актуальной становится задача создания программных комплексов обеспечивающих возможность обработки данных в компьютерной сети, а так же удаленной обработки через глобальную сеть Интернет. Поскольку на сегодняшний день широко используются различные операционные

системы (Windows, операционные системы с открытым кодом на базе платформы Unix, Mac OS), то при разработке удаленной системы анализа данных приходится заботиться об обеспечении переносимости разрабатываемой системы на различные платформы.

В качестве инструментария для решения перечисленных выше задач были использованы технологии на базе популярного языка программирования Java. Общую схему всей удаленной системы анализа данных можно представить как набор, состоящий из сервера и компьютеров, подключаемых к нему по сетевым протоколам (рис. 1), на которых установлены части системы, отвечающие за непосредственную работу с пользователями (клиенты) и проведение анализа данных (обработчики).



Рис. 1. Общая схема удаленной системы

В самом начале разработки системы уже существовал большой набор алгоритмов для проведения анализа данных, доступных с использованием технологии Common Object Model (COM). По этой причине часть разрабатываемого программного комплекса, непосредственно связанная с анализом данных, основана на использовании существующих COM-объектов приложением на языке Java, со своей стороны обеспечивающим платформенную независимость и возможность использования обработчика для проведения удаленных вычислений (рис. 2).

Задача взаимодействия объектов Java с COM объектами C++ была решена при помощи технологии Java Native Interfaces (JNI), а в качестве модели обмена данными между объектами была использована Document Object Model (DOM).

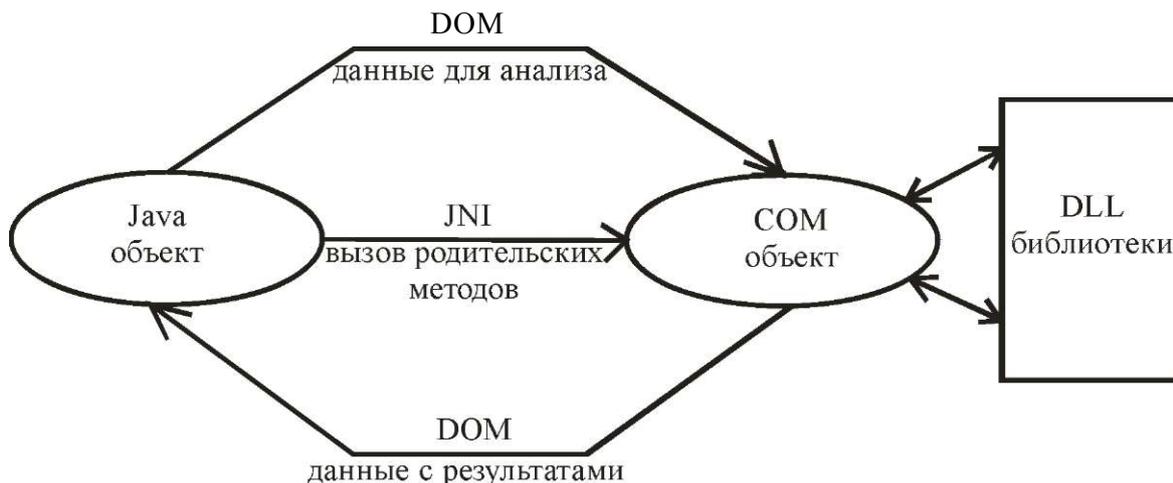


Рис. 2. Общая схема обработчика данных

Сам по себе обработчик представляет Java-приложение, написанное с использованием технологии `java.net` [3], которое время от времени отправляет запросы на сервер, содержащие информацию об IP адресе данного компьютера и номере порта, через который будет осуществляться подключение к данному обработчику. На стороне обработчика организован мини-интерфейс с использованием библиотеки `SWING` [4], где можно указать адрес сценария или сервлета, отвечающего за подключение к серверу, и просмотреть историю подключений и ошибок.

После начала работы обработчик переходит в режим ожидания. При получении данных от сервера и запроса на их обработку на стороне обработчика автоматически создаются Java объекты, необходимые для анализа полученных данных, а после окончания обработки результаты отправляются обратно на сервер. Отличительной особенностью созданного приложения-обработчика является то, что каждый запрос обрабатывается в отдельном потоке, что дает возможность одновременно анализировать сразу множество данных на одном обработчике. Обмен данными между обработчиком и серверной частью происходит на уровне протоколов в виде потоков байтов. Поэтому не имеет значения, на каком программном языке реализована серверная часть программного комплекса, и под управлением какой операционной системы она работает.

Серверная часть созданного программного комплекса (рис. 3) основана на технологии `Java Servlets` [5]. Основная задача серверной части состоит в том, чтобы при каждом запросе со стороны клиента отыскать свободный обработчик, отправить ему данные и отослать ответ клиенту обратно. Кроме отправки результатов проведенного анализа,

клиенту так же отправляется информация о любых исключениях, возникших в результате обработки. Таким образом, клиент всегда владеет информацией о результате проведенных операций.

запрос

Web - сервер:
*выбор калькулятора;
обработка запросов;
формирование ответа.*

Калькулятор 1

ОТВЕТ

Калькулятор n

Рис. 3. Работа серверной части

Разработанная система поддерживает удаленную модернизацию серверной части с использованием скриптов приложения Ant.

Последним этапом разработки удаленной системы было создание пользовательского интерфейса клиентской части системы (рис. 4).

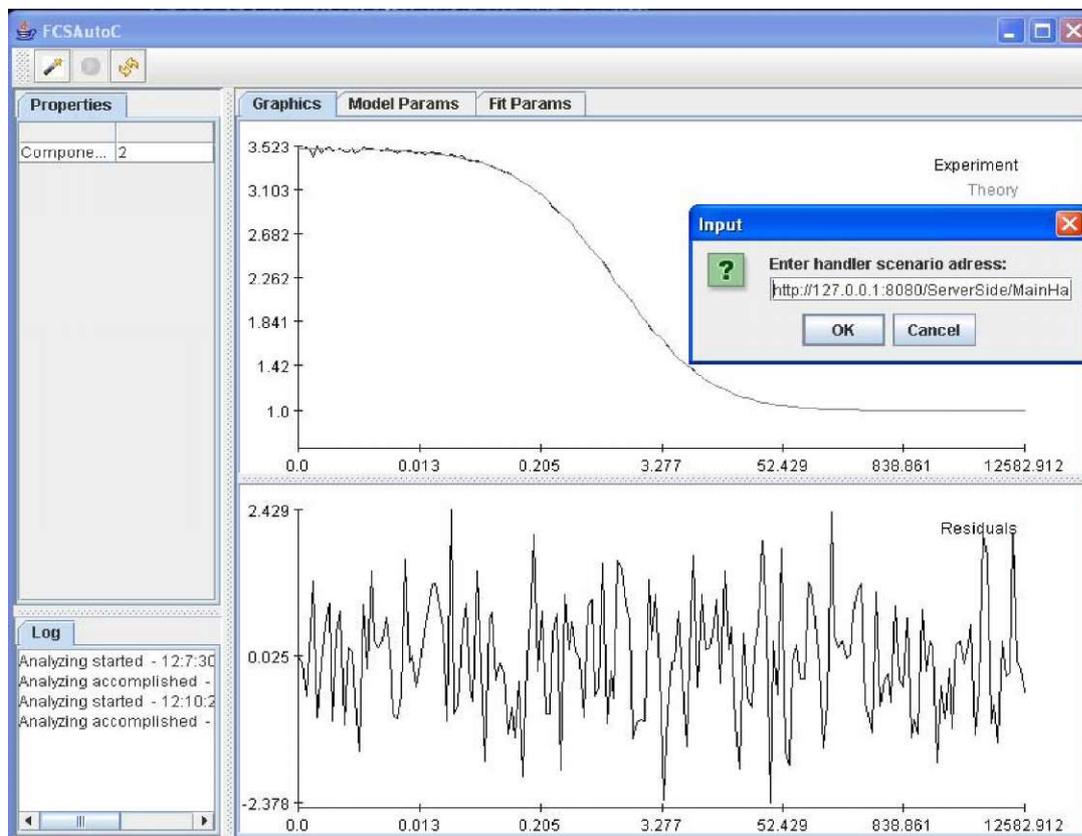


Рис. 4. Общий вид пользовательского клиентского интерфейса

С его помощью пользователь имеет возможность быстро сконфигурировать данные для обработки перед посылкой на сервер, выбрать экспериментальные данные, задать начальные приближения для оцениваемых параметров, указать адрес сценария или сервлета, ответственного за прием и обработку данных, а также просмотреть результаты проведенного анализа.

Как и в случае взаимодействия обработчиков с сервером, обмен данными между клиентами и сервером происходит на уровне протоколов в виде потоков байтов, что делает возможным использовать различные сервера, такие, например, как Tomcat или Internet Information Server компании Microsoft.

Таким образом, разработанный программный комплекс обеспечивает возможность подготовки исходных данных флуоресцентных экспериментов и схемы их анализа, передачи исходной информации для удаленной обработки, а так же получение и отображение результатов проведенных вычислений. Созданное программное обеспечение дает возможность одновременной обработки различных наборов данных на отдельных обработчиках, доступных серверной части. Обмен данными между отдельными частями системы обработки данных организован на уровне сетевых протоколов, что делает ее независимой от программной реализации серверной части. Применение Java-технологий для создания клиентской части программного комплекса дает возможность ее использования в неизменном виде на различных операционных системах.

Литература

1. *Лакович, Д.* Основы флуоресцентной спектроскопии. М. : Мир. 1986.
2. *Yatskou, M. M.* Integrated data analysis in time-resolved fluorescence and fluorescence correlation spectroscopy / M. M. Yatskou, A. V. Digris, E. G. Novikov, V. V. Skakun, V. V. Aranasovich // Recent Res. Devel. in Physical Chem. Vol. 7. Part I. 2004. P. 165-183.
3. *Хортсман, К. С.* Java 2. Библиотека профессионала, том II. Тонкости программирования, 7-е изд. / Пер. с англ. М. : Издательский дом «Вильямс». 2007. С. 189-235.
4. *Портянкин, И.* Библиотека программиста. SWING эффективные пользовательские интерфейсы. Питер. 2005.
5. *Пери, Б.* Java сервлеты и JSP: сборник рецептов. 2-е изд. // Пер. с англ. М. : КУДИЦ-ПРЕСС. 2006.