

УДК 004.91+347.78.031

ОСОБЕННОСТИ АРХИТЕКТУРНОЙ РЕАЛИЗАЦИИ СИСТЕМЫ ВИЗУАЛИЗАЦИИ И ОБРАБОТКИ РЕЗУЛЬТАТОВ СПЕКТРАЛЬНОГО АНАЛИЗА

Л.В. РУДИКОВА

Гродненский государственный университет имени Янки Купалы
Ожешко, 22, Гродно, 230023, Беларусь

Поступила в редакцию 2 октября 2014

Приводятся особенности архитектурной реализации системы, предназначенной для визуализации и обработки результатов спектрального анализа. Архитектурой системы предусмотрено наличие клиентов двух типов: десктопный клиент и клиентское приложение веб-браузера. Разработанное по предлагаемым принципам программное обеспечение позволит получать быстрые и точные результаты экспертизы, накапливать результаты спектрального анализа в базу данных для их последующей обработки, а также визуализировать, масштабировать и детализировать полученные спектры.

Ключевые слова: спектральный анализ, архитектура реализации, клиентское десктоп-приложение, приложение веб-браузера, база данных, сервисное приложение.

Введение

При рассмотрении проблем, которые возникают при проведении лазерного спектрального микроанализа, можно выделить проблему идентификации и обработки эмиссионных спектров лазерной абляционной плазмы. Следует отметить, что некоторые особенности лазерной плазмы не позволяют полностью автоматизировать обработку спектров. В силу этого требуется тесное взаимодействие человека с программой, при помощи которой, спектр обрабатывается. Наличие графического интерфейса существенно упрощает и ускоряет работу с программой, а также делает ее интуитивно более понятной [1, 2].

Процесс обработки спектра можно разделить на несколько основных этапов.

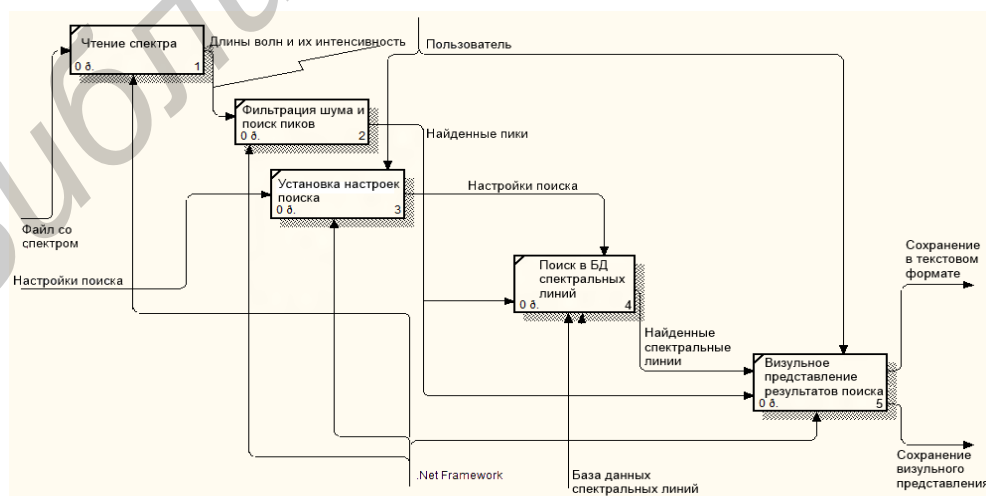


Рис. 1. Процесс обработки спектра

Обработка спектра

Основное место в программном обеспечении, связанном с обработкой и визуализацией спектров, занимает поиск пиков. Отметим, что алгоритм поиска пиков основан на методе деконволюции сигналов и осуществляется в несколько этапов.

1. Вычисление фона сигнала. Результатом является информация, содержащая пики и фон. Данный метод основан на SNIP-алгоритме (Sensitive Nonlinear Iterative Peak):

$$V_p(i) = \min \left\{ V_{p-1}(i), \frac{[V_{p-1}(i+p) + V_{p-1}(i-p)]}{2} \right\}, \quad (1)$$

где p – номер итерации.

2. Вычисление сглаженного сигнала с помощью алгоритма, который базируется на дискретной цепи Маркова, которая имеет простое инвариантное распределение:

$$U_2 = \frac{P_{1,2}}{P_{2,1}} U_1, U_3 = \frac{P_{1,2}}{P_{2,1}} U_2 U_1, \dots, U_n = \frac{P_{n-1,n}}{P_{n,n-1}} U_{n-1} \dots U_2 U_1, \quad (2)$$

где U_1 – определяется из нормализованного условия (3):

$$\sum_{i=1}^n U_1 = 1. \quad (3)$$

3. Вероятность изменения позиции пика от канала i к каналу $i+1$:

$$P_{i,i+1} = A_i \sum_{k=1}^m \exp \left(\frac{y(i \pm k) - y(i)}{y(i \pm k) + y(i)} \right), \quad (4)$$

где A_i – такая константа, что выполняется равенство:

$$P_{i,i-1} + P_{i,i+1} = 1. \quad (5)$$

4. Вычисляется вектор функции g , необходимой для решения уравнения свертки (6) (с учетом параметра функции поиска пиков):

$$f * g = h, \quad (6)$$

где h – записанный сигнал; f – восстановленный сигнал.

g имеет вид:

$$G(i) = 1000 * e^{-\frac{[i-3*s]^2}{2*s^2}}, \quad (7)$$

где s – параметр функции поиска пиков.

Поиск пиков производится на восстановленном сигнале с учетом порогового значения, указанного в процентах. Пороговое значение высчитывается из значения наибольшего пика и означает, что пики, имеющие значение меньше порогового, не учитываются.

Для вычисления высоты и площади найденных пиков используется уровень фона, полученный ранее на первом этапе.

Общая архитектура системы, связанной с визуализацией и обработкой результатов спектрального анализа

Для программного обеспечения, связанного с визуализацией и обработкой результатов спектрального анализа, была разработана 4-х уровневая архитектура, представленная на рис. 2.

Архитектурой системы предусмотрено наличие клиентов двух типов: десктопный клиент и клиентское приложение веб-браузера.

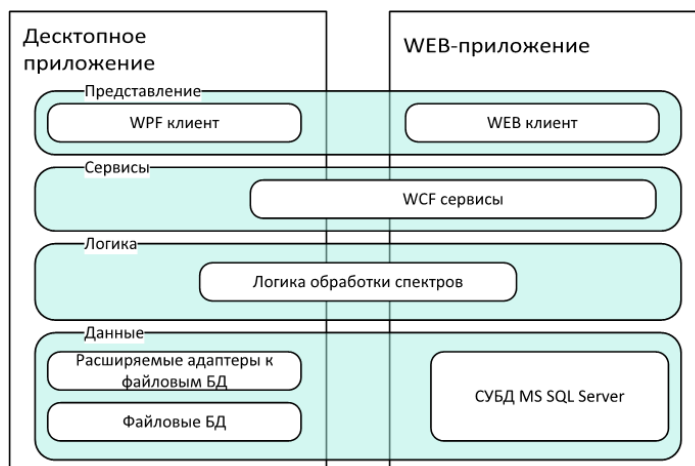


Рис. 2. Четырехуровневая архитектура универсальной системы, связанной с визуализацией и обработкой результатов спектрального анализа

Для централизованного доступа к системе предусмотрено веб-приложение, посредством которого могут быть реализованы следующие возможности.

1. Получение информации о типе организации, которой принадлежит данная система, типе исследуемых объектов и другой информации об организации.
2. Скачивание части библиотеки спектральных линий, собранной на основе исследований данной организации, если предусмотрен публичный доступ.
3. Предоставление веб-интерфейса для программирования. Данный API предоставляет возможности работы по поиску спектральных линий и функционал экспертных оценок, основанный на открытой части библиотеки спектральных линий данной организации, если таковая имеется.

Веб-приложение использует сервисы для доступа к бизнес-логике и использует реляционную БД MS SQL для хранения библиотек спектральных линий.

Для индивидуального пользования данной системой предусмотрено десктоп-приложение, которое предоставляет следующие возможности.

1. Загрузка и графическое отображение результатов измерений.
2. Поиск пиков и соответствующих спектральных линий в локальных файловых базах данных либо в централизованной БД.
3. Экспорт результатов работы в файл, изображение либо сохранение результатов в централизованную библиотеку спектральных линий.

Особенностью архитектуры десктоп-приложения является возможность доступа, как к уровню сервисов, так и к уровню бизнес-логики. Это позволяет легко преобразовать индивидуальное использование программы в централизованное. Для этого достаточно развернуть сервер базы данных и сервисы системы в интрасети.

Указанная особенность предполагает возможность хранения информации как в файлах на локальном компьютере, так и в централизованной библиотеке, основанной на реляционной базе данных MS SQL. Это позволяет, например, нескольким подразделениям в организации использовать соответствующие непересекающиеся хранилища информации, находящиеся в интрасети, а также составлять библиотеки спектральных линий для конкретного типа исследуемых объектов.

Общая архитектура реализации системы, связанной с визуализацией и обработкой результатов спектрального анализа

Общая архитектура реализации системы визуализации и обработки результатов спектрального анализа представлена на рис. 3. Система представляется в виде трех уровней,

имеющих минимальные связи между собой. Перечислим уровни, составляющие систему: база данных (Database); Back-end, представленный сервисным приложением, и пользовательское приложение (Front-end).

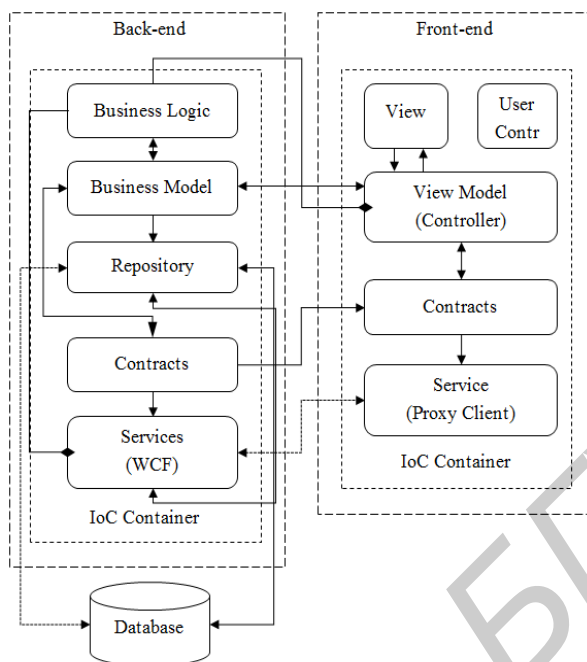


Рис. 3. Общая архитектура реализации

Для разработки предлагаемого программного обеспечения, поддерживающего лазерную экспрессную экспертизу, выбраны следующие технологии и инструменты разработки: платформа .Net Framework; объектно-ориентированный язык C#; СУБД MS SQL Server; Microsoft Enterprise Library.

Рассмотрим выделенные уровни и их компоненты более подробно.

База данных (Database). Данный уровень представляет собой отдельный сервер, на котором развернута СУБД. Уровень базы данных связан с сервисным проецированием бизнес-моделей на реляционные сущности БД с помощью репозитория (компонент Repository). Репозиторий также ответственен за работу с данными (сохранение, обновление, удаление, выборка). Основой репозитория является ADO.NET Entity Framework технология. Данная технология позволяет незаметно для разработчика преобразовывать LINQ-запросы и другие выражения в эквивалентные SQL-запросы и обратно – результаты выполнения запросов к объектам бизнес сущностей.

Сервисное приложение (Back-end). Сервисное приложение представлено WCF-сервисами. Для уменьшения зависимостей между компонентами данного уровня используется IoC-контейнер.

Уровень сервисного приложения включает в себя следующие компоненты.

1. Бизнес-логика (Business Logic). Реализация логики по работе со спектрами, логика экспертной системы, логика работы с бизнес-сущностями и др.

2. Бизнес-модели (Business Model). Определены сущности, отражающие предметную область в виде объектов системы.

3. Репозиторий (Repository). Реализация логики по работе с данными, формирование запросов к БД, проецирование бизнес-моделей на сущности БД, преобразование запросов в эквивалентные SQL-запросы и обратное преобразование результатов выполнения запросов к объектам бизнес-сущностей с использованием технологии ADO.NET Entity Framework, выбор провайдеров, в зависимости от типа БД.

4. Контракты (Contracts). Описание операций, реализуемых сервисом.

5. Сервисы (WCF Services). Реализация контрактов, предоставление конечных точек сервиса, определение протоколов, адресов, типов шифрования передачи данных и др.

Пользовательское приложение (Front-end). Уровень пользовательского приложения имеет зависимость от сервисного приложения в виде контрактов, реализованных сервисным приложением. Пользовательское приложение также включает компонент бизнес-моделей, для отображения информации, запрошенной у сервисного приложения.

Уровень пользовательского приложения включает в себя следующие компоненты: пользовательское представление (View); пользовательские элементы управления (User Controls); модель представления (View Model); контракты (Contracts) и сервисы (Service Proxy Client). Пользовательское приложение реализовано в виде двух клиентов: десктопного клиента и клиентского приложения веб-браузера.

Главное назначение десктопного клиента – быстро обрабатывать зарегистрированные спектры и предоставлять необходимые данные в удобном для пользователя виде (рис. 4).

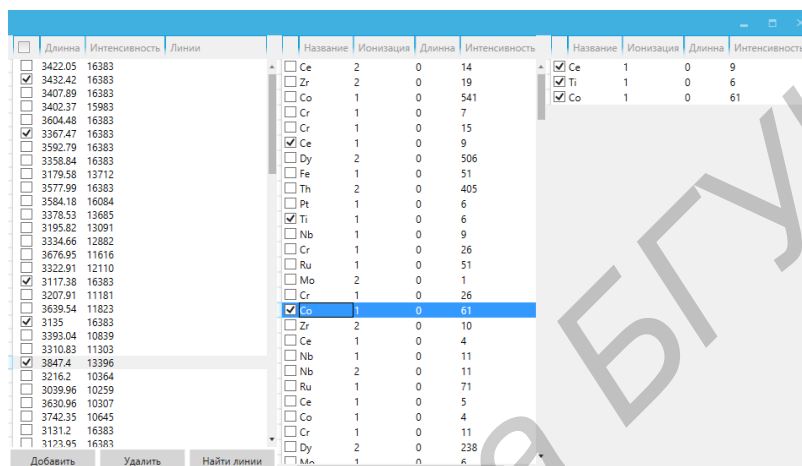


Рис. 4. Клиентское десктоп-приложение: окно задания параметров поиска спектральных линий

Разработанное приложение обеспечивает следующие возможности: считывание файлов, содержащих зарегистрированные спектры, группировка и сохранение их в истории; поиск пиков с возможностью отсека пиков по высоте и ширине; возможность настройки параметров шумоподавления для более точного определения границ пиков; возможность выбора различных баз данных (на данный момент NIST и max форматы) и поиск спектральных линий с возможностью настройки поиска; сохранение проделанной работы в виде графического изображения с подписанными линиями либо в базу данных, содержащую накопленные результаты работы других пользователей посредством сервисов.

Для уменьшения зависимостей между компонентами используется Unity Application Block (Unity). Unity реализует паттерн Dependency Injection (Внедрение зависимости). Таким образом, клиентское десктоп приложение зависит только от интерфейсов (контрактов), а не от самой реализации. Благодаря этому данное приложение легко расширяемо, даже сторонними модулями.

В разработанном приложении имеется две точки расширения функциональности: чтение файлов, содержащих зарегистрированные спектры и подключение различных баз данных спектральных линий. Расширение функциональности доступно посредством реализации определенных интерфейсов (ISpectrumFileReader (листинг 1) и ISpectrumLinesRepository (листинг 2), соответственно), которые доступны в динамически загружаемых библиотеках SpectrumAnalyser.Data.Files.SpectrumFileReaderContracts.dll и SpectrumAnalyser.Data.Files.SpectrumDatabaseContracts.dll соответственно, и подключением реализованных модулей (модули должны находиться в одной папке с приложением и быть названы SpectrumAnalyser.Data.Files.*.dll). Модули расширения функционала должны быть написаны на языках семейства .Net 4.0.

Листинг 1. Описание интерфейса ISpectrumFileReader

```
public interface ISpectrumFileReader
{
    SpectrumFile ReadSpectrumFile(FileInfo fileInfo);
    string SpectrumFileType { get; }
    IList<string> AcceptableExtensions { get; }
}
```

Листинг 2. Описание интерфейса ISpectrumLinesRepository

```
public interface ISpectrumLinesRepository
{
    IList<Peak> FillMatchedPeaks(IList<Peak> peaks, SpectrumDbConnection
    connection, LinesSearchSettings linesSearchSettings);
    string ConnectionType { get; }
}
```

Свойства ConnectionType и SpectrumFileType должны быть уникальными в пределах всех реализаций соответствующих классов.

Данные модули загружаются при запуске приложения и регистрируются в Unity Container. Пример загрузки модулей поиска линий в спектральных базах приведен в листинге 3.

Листинг 3. Загрузка модулей поиска линий в файловых БД

```
var currentLocation = Directory.GetCurrentDirectory();
var repositoryFiles =
    Directory.GetFiles(currentLocation, "SpectrumAnalyser.Data.Files.*.dll");
foreach (var repositoryFile in repositoryFiles)
{
    Assembly assembly = Assembly.LoadFile(repositoryFile);
    var allTypes = assembly.GetTypes().ToList();
    foreach (var allType in allTypes)
    {
        var fileReaderTypes = allType.GetInterfaces().Where(x =>
            x.GetTypeInfo().AsType() == typeof(ISpectrumFileReader));
        if (fileReaderTypes.Any())
        {
            var spectrumDataBaseInstance =
                (ISpectrumFileReader)Activator.CreateInstance(allType);
            UnityContainer.RegisterInstance(typeof (ISpectrumFileReader),
                spectrumDataBaseInstance.SpectrumFileType,
                spectrumDataBaseInstance);
        }
        var spectrumDataBaseTypes = allType.GetInterfaces().Where(x =>
            x.GetTypeInfo().AsType() == typeof (ISpectrumLinesRepository));
        if (spectrumDataBaseTypes.Any())
        {
            var spectrumDataBaseInstance = (ISpectrumLinesRepository)
                Activator.CreateInstance(allType);
            UnityContainer.RegisterInstance(typeof
                (ISpectrumLinesRepository),
                spectrumDataBaseInstance.ConnectionType
                spectrumDataBaseInstance);
        }
    }
}
```

После того как модули были зарегистрированы в Unity Container, их можно использовать в любой части кода, где доступен UnityContainer. Так как загрузки модулей поиска линий в спектральных базах регистрируются несколько раз по одному интерфейсу, то был введен уникальный параметр для данных модулей – тип базы. Одновременно в Unity Container может находиться только один модуль определенного типа базы данных.

Клиентское приложение веб-браузера реализовано с использованием технологии ХВАР (XAML Browser Application) и представляет собой полноценное WPF-приложение с некоторыми особенностями:

– приложение может быть встроено в HTML-страницу (с помощью тега <iframe>) либо загружено по прямому адресу и занимать всю область веб-страницы;

- приложение имеет ограниченные права в целях предотвращения выполнения потенциально опасного кода;
- приложение не требует инсталляции – при первой загрузке приложение кэшируется, но если приложение было обновлено на сервере, то на клиентской машине приложение кэшируется заново, таким образом клиент всегда работает с последней версией приложения;
- приложение может быть запущено только в браузерах Internet Explorer или Firefox;
- приложение имеет черты как веб-приложения (должно быть развернуто на сервере), так и многофункционального десктоп-приложения (использование преимуществ WPF).

Аналогично клиентскому десктоп-приложению, назначением приложения веб-браузера является быстрая обработка зарегистрированных спектров и предоставление данных в удобном для пользователя виде. Однако в сравнении с десктоп-приложением веб-приложение имеет ряд отличий. Во-первых, в качестве базы данных спектральных линий используется обобщенная библиотека спектральных линий (в десктоп-приложении возможно использование файловых БД). Во-вторых, конфигурация приложения хранится в базе данных (в десктоп-приложении используется конфигурационный файл). Приложение веб-браузера и десктоп-приложение имеют общую кодовую базу (пользовательские элементы управления, модели представления, конвертеры и др.), которая вынесена в отдельный проект. Отличием является запуск приложения, т.к. он производится в веб-браузере, а не в окне.

Заключение

Приведены особенности архитектурной реализации системы, предназначенной для визуализации и обработки результатов спектрального анализа. Разработанное по предлагаемым архитектурным решениям программное обеспечение позволяет получать быстрые и точные результаты экспертизы, накапливать результаты спектрального анализа в базу данных для их последующей обработки, а также визуализировать, масштабировать и детализировать полученные спектры. Несомненно, предлагаемая разработка является актуальной и будет востребована при проведении экспертиз широкого круга объектов как в различных отраслях народного хозяйства Республики Беларусь, так и в научных исследованиях.

Результаты работы получены в процессе выполнения ГПНИ «Разработка научно-методического обеспечения практического использования мобильных лазерных спектроаналитических систем и рентгенофлуоресцентного анализатора для экспрессной материаловедческой экспертизы в инновационных технологиях, предотвращении чрезвычайных ситуаций, экологии, криминалистике, сохранении историко-художественного наследия. Разработка и адаптация программного обеспечения для использования при проведении экспрессной материаловедческой экспертизы различных изделий и образцов».

ARCHITECTURAL FEATURES OF THE IMAGING AND TREATMENT IMPLEMENTATION OF SPECTRAL ANALYSIS RESULTS

L.V. RUDIKOVA

Abstract

The architectural features of the implementation of a system for processing and visualization of results of spectral analysis is described. Architecture provides for a system of two types of clients: a desktop client and the client application is a web browser. Developed on the proposed principles software can afford to get fast and accurate results of the examination, to accumulate the results of spectral analysis in the database for further processing, as well as visualize, scale and detail the spectra.

Список литературы

1. Рудикова Л.В. // Матер. IX Междунар. науч. конф. «Лазерная физика и оптические технологии». Гродно, 30 мая–2 июня 2012 г. С. 115–118.
2. Рудикова Л.В. // Докл. БГУИР. 2013. №3 (73). С. 26–32.