

СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ

УДК 004.422.81

УНИФИЦИРОВАННАЯ ПРОГРАММНАЯ ПЛАТФОРМА СОЗДАНИЯ СРЕДСТВ ОБРАБОТКИ ДАННЫХ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ ЗЕМЛИ - «ER-SET»

В. И. Побаруев, в.н.с. НИИ «Фотон» РГРТУ, к.т.н., доцент; foton@rsreu.ru

*Рассматриваются архитектура, принципы функционирования, способы программирования и технические характеристики программной платформы, которая разработана в НИИ «Фотон» и применяется для создания специализированного программного обеспечения обработки данных от систем дистанционного зондирования Земли. **Целью работы** является создание технологии быстрой разработки высокопроизводительных приложений для обработки данных дистанционного зондирования Земли. Рассматриваемая программная платформа представляет собой набор компонент (шаблонов и модулей), бинарных интерфейсов и правил интеграции элементов в единую систему. Высокая производительность создаваемых на основе платформы программных комплексов обеспечивается благодаря алгоритмическим и архитектурным решениям. Технология быстрой сборки приложений основана на глубокой унификации разрабатываемых программных средств (DLL-модулей), интерфейсов и правил их интеграции.*

Ключевые слова: дистанционное зондирование, программная система, методы программирования, интерфейс, производительность, архитектура программного обеспечения, быстрая разработка приложений, унификация.

Введение

Ежегодно в НИИ «Фотон» при Рязанском государственном радиотехническом университете разрабатываются и внедряются в эксплуатацию несколько десятков специализированных программных комплексов (ПК), предназначенных для обработки данных дистанционного зондирования Земли от отечественных и от зарубежных систем. Программные комплексы, находящиеся в эксплуатации, постоянно модифицируются, наращиваются их функциональные возможности и разрабатываются новые версии. Каждый комплекс – это миллионы строк программного кода (для сравнения, 1 миллион строк – это порядка 18000 страниц текста книги). Естественно, что выполнять ежегодно такой объем работ было бы невозможно, если бы не применялись специальные технологии программирования.

Программные технологии, позволяющие выполнять быструю разработку приложений, появились в НИИ «Фотон» не сразу. Первая вер-

сия программной платформы появилась в 1998 году с созданием ПК «NormSat». Данный ПК создавался по заказу Российского НИИ Космического приборостроения (ныне АО «Российские космические системы»). На тот момент это был единственный программный комплекс, который разрабатывался в НИИ «Фотон» (в то время подразделение называлось «Лаборатория обработки аэрокосмических изображений»). С течением времени развитие платформы позволило разрабатывать не один ПК в год, а несколько (при том же численном составе сотрудников). В результате выросли финансовые возможности НИИ «Фотон», что позволило расширить штат сотрудников и еще больше увеличить объемы выполняемых работ.

Основное назначение программных комплексов, создаваемых на основе представленной платформы, – это обработка изображений (оптических, инфракрасных, радиолокационных), которые представляют собой сверхбольшие массивы информации (сотни гигабайт). Разработанная

программная платформа и применяемые технологии ее программирования позволяют увеличить производительность труда программистов, создающих программные комплексы для обработки данных дистанционного зондирования Земли, в десятки раз, по сравнению с созданием ПК традиционными методами. По сути, платформа представляет собой конструктор программных комплексов и частично реализует функции операционной системы (ОС), в рамках которой и согласно ее внутренним правилам реализуются комплексы обработки. Данная программная платформа запатентована [1] и имеет название «ER-Set». Более ранние публикации по данной теме раскрывали лишь отдельные элементы представленной технологии [2-5]. В данной статье впервые изложено интегрированное описание ключевых программно-технических решений программной платформы «ER-Set», представлены ее архитектура и схема функционирования в целом.

1. Архитектура платформы

Платформа «ER-Set» представляет собой совокупность компонент, программных средств и интерфейсов, реализующих технологию быстрого создания приложений. Архитектура платформы строится на следующих принципах:

- унификации средств доступа к растровым и атрибутивным данным, за счет чего обеспечивается универсальность применения разработанных DLL-модулей и многократное их использование в различных проектах;
- организации собственной высокопроизводительной подсистемы виртуальной памяти и межзадачного обмена информацией, учитывающей специфику обработки данных ДЗЗ;

- быстрого и легкого в программировании межпрограммного интерфейса, обеспечивающего взаимодействие компонент системы, реализованных в различных средах разработки;

- виртуальной организации вычислительного процесса;

- стандартизации пользовательского интерфейса и способа сборки готовых приложений.

Типичная структура программного комплекса, разработанного на основе платформы «ER-Set» представлена на рисунке 1.

Основу платформы составляет «ассоциативный» интерфейс межпрограммного взаимодействия компонент [3]. Интерфейс представляет собой набор функций, классов и данных, которые предназначены для обеспечения взаимодействия компонентов системы между собой. Управление интерфейсом межпрограммного взаимодействия и компонентами осуществляется в ядре (в центральном управляющем модуле). Интерфейс по своей структуре является однородным, открытым и общим как для ядра, так и для DLL-модулей. Любой DLL-модуль может расширить интерфейс собственными функциями или данными. Взаимодействие компонент комплекса с внешними файлами изображений производится посредством модулей импорта/экспорта. Данные модули обеспечивают преобразование информации из внешней структуры (файла) во внутреннее унифицированное представление и обратно. Все остальные модули работают только с унифицированным (внутренним) представлением данных. Благодаря такому подходу обеспечивается возможность повторного использования как DLL-модулей, так и отдельных фрагментов ранее разработанного программного кода вне зависимости от формата представления исходной информации.



Рисунок 1 – Архитектура программного комплекса на основе платформы «ER-Set»

Программный комплекс, разрабатываемый на основе платформы «ER-Set», можно условно разделить на 2 сегмента. Первый сегмент (А) представляет собой объекто-зависимую часть, то есть модули и функции, являющиеся уникальными для конкретной системы ДЗЗ. Второй сегмент (В) – это набор универсального программного обеспечения, инвариантного к форматам входных данных. Этот сегмент является наиболее объемным по количеству модулей и реализуемых функций.

Разработка модулей из второго сегмента в большинстве случаев не требуется (то есть при разработке новых комплексов данное программное обеспечение используется повторно в виде DLL-модулей без необходимости их перекомпиляции). Такой подход при проектировании систем обработки данных ДЗЗ для вновь создаваемых КА позволяет осуществлять разработку ПК в сжатые сроки посредством использования ранее разработанных модулей для решения типичных задач и выделить достаточные ресурсы на решение предметной задачи.

2. Организация виртуальной памяти и внутреннее представление объектов изображений

Программные системы ДЗЗ предназначены для обработки данных сверхбольшого объема (обычно десятки и сотни гигабайт). В этой связи высокая производительность программного комплекса может быть обеспечена только путем организации эффективного видеоинформационного обмена. С этой целью в ядре системы «ER-Set» разработан гибридный механизм кэширования данных изображений в памяти. По сути, платформа «ER-Set» представляет собой менеджер виртуальной памяти с характеристиками и сервисом, присущим операционным системам, но учитывающим специфику обработки данных ДЗЗ. Система использует для своих нужд всю оперативную память компьютера (для управления оперативной памятью используются низкоуровневые сервисы ОС) и сама распределяет ее между задачами. Внутренний кэш автоматически представляет многомерную растровую матрицу изображения в виде блочной пирамиды (по типу вейвлет-преобразования). Благодаря такому подходу обеспечивается очень высокая производительность программных комплексов, создаваемых на данной платформе, по сравнению с конкурирующими решениями (при этом не расходуется дополнительная оперативная память на пирамидальное представление, то есть применяется хэш-пирамида [4]). Высокая скорость обработки разработанного на основе «ER-Set» про-

граммного обеспечения (до настоящего времени недостижимая ни для одной из программ аналогичного назначения) подтверждается отзывами эксплуатирующих организаций, таких как НЦ ОМЗ, НИЦ Планета, УП Геоинформационные системы Республики Беларусь и др.

Программный доступ к растровым данным построен таким образом, чтобы абстрагироваться от физического формата хранения матрицы изображения. В процедурах обработки информация о формате раstra может быть использована только как вспомогательная, поскольку строка матрицы изображения может быть запрошена в том формате, на который настроен DLL-модуль. Например, если процедура обработки запрашивает матрицу изображения в вещественном формате, то вне зависимости от физического формата хранения исходного изображения строка матрицы будет прочитана и обработана в вещественном формате. Так реализуются универсальность (унификация) программного комплекса и инвариантность модулей обработки по отношению к различным типам и форматам входных данных.

Доступ к матрице изображений в системе «ER-Set» осуществляется построчно и единообразно с помощью единой функции, прототип которой на языке C++ представлен ниже.

```
DWORD ReadDataStream (HIMAGE
hImage,void * pBuffer,long dwLine,DWORD
ReadFormat=0, long StartSample=-1,long
RBlockSize=-1,DWORD CacheAloritm=0);
```

где hImage – дескриптор объекта изображения;

pBuffer – буфер для данных строки изображения;

dwLine – номер читаемой строки,

ReadFormat – запрашиваемый формат данных,

StartSample – первый читаемый элемент,

RBlockSize – размер блока данных,

CacheAloritm – номер алгоритма кэширования.

Матрицы изображений в системе представляются в виде многомерных и нетипизованных растров (то есть могут быть запрошены в разных форматах, например байтами, словами, числами с плавающей точкой, числами с фиксированной точкой и т.д.).

Внутренний способ хранения матрицы изображения и ее физическое расположение (в памяти или на внешнем носителе) прозрачны для программиста. Для внутреннего хранения применяется хэш-тейловый метод организации данных изображения [4].

Для пояснения идеи хэш-тейловой организации рассмотрим вначале принцип тейловой (фрагментарной) организации, который показан на рисунке 2,а. Экспериментальные размеры тейла следующие: количество строк в тейле 16, длина подстроки 4096 байт. Таким образом, например, если первая строка одноканального (одномерного) изображения имеет длину 36000 пикселей и радиометрическое разрешение 16 бит/пиксель, то она будет располагаться в первых строках тейлов с номерами 0-17. При этом количество элементов в строке тейла зависит от радиометрического разрешения снимка (разрядности) и числа спектральных каналов (слоев).

Рассмотрим тейловую организацию памяти в сравнении с хэш-тейловой организацией (рисунки 2).

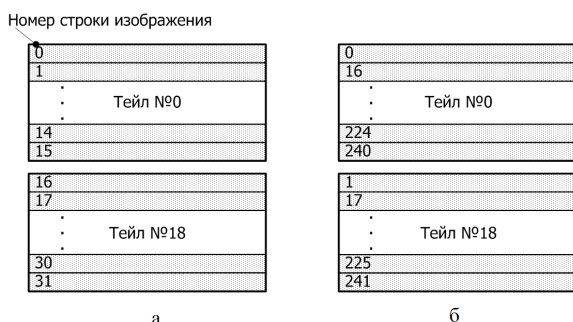


Рисунок 2 – Расположение строк изображения в тейлах: а – тейловая организация, б – хэш-тейловая организация

Из рисунка 2,б видно, что при хэш-тейловой организации тейлы формируются из строк с межстрочным интервалом $S=16$, который подобран эмпирически.

Преимущество такой организации матриц изображения проявляется при работе с данными сверхбольшого объема, когда объем данных значительно превышает ресурсы оперативной памяти. Например, для отображения изображения на экране монитора в масштабе 1:16 необходимо считать только один из 16 тейлов данных. По сути, матрицы имеют пирамидальное представление, но при этом на формирование пирамидальных слоев не затрачиваются дополнительные ресурсы памяти. При обработке и отображении данных в уменьшенном масштабе, за счет выборки (размещения в оперативной памяти) меньшего количества тейлов, данная организация матриц получает выигрыш, кратный масштабу. Если производится обработка всего объема данных (в масштабе 1:1), то скорости обмена данными в тейловом и хэш-тейловом подходах будут одинаковы.

3. Организация межпрограммного интерфейса

Спроектировать сложную программную систему можно только при грамотном решении вопросов, связанных с декомпозицией, то есть разложением единой системы на отдельные части или компоненты. В этой связи одним из наиболее важных аспектов проектирования сложных средств обработки является способ межпрограммного взаимодействия отдельных компонентов (интерфейс). Интерфейс межпрограммного взаимодействия определяет такие характеристики системы обработки, как:

- гибкость (адаптируемость) к различным типам задач и данных;
- производительность;
- возможность организации повторного и многократного использования компонентов;
- простота освоения, что обеспечивает возможность быстрой разработки приложений.

В «ER-Set» разработан собственный протокол межмодульного взаимодействия, который получил название «Ассоциативный интерфейс» (Associative Interface - AI) [3]. Он основан на получении доступа к интерфейсу (совокупности методов и данных для взаимодействия) посредством уникальных имен, в качестве которых выступают уникальные текстовые строки (аналог GUID, используемый в COM [6]). Но, в отличие от COM в AI используются не совокупности методов (которые могут быть реализованы с помощью классов, например на языке C++), а отдельные элементы – указатели на функции и структуры данных. На основе этих отдельных компонентов формируются классы интерфейса применительно для конкретного языка программирования, то есть единый программный код, включаемый во все модули системы и обеспечивающий их взаимодействие (рисунок 3).

Благодаря этому при добавлении нового элемента (функции или структуры данных) в интерфейс он автоматически расширяется без дополнительных изменений. То есть в интерфейсный класс просто добавляется новый метод (в COM же в этом случае требуется создание нового интерфейса), посредством чего реализуется наследование. Для использования нового метода в интерфейсе требуется перекомпиляция модуля, однако остальные модули со «старым» вариантом интерфейса остаются полностью совместимы и работоспособны.

Интерфейс представляет собой набор адресов функций и структур данных, каждому из которых присвоено уникальное имя. В программной системе должен присутствовать модуль, ко-

торый обеспечивает управление ассоциированными адресами (содержит таблицу имен и обеспечивает их управление).

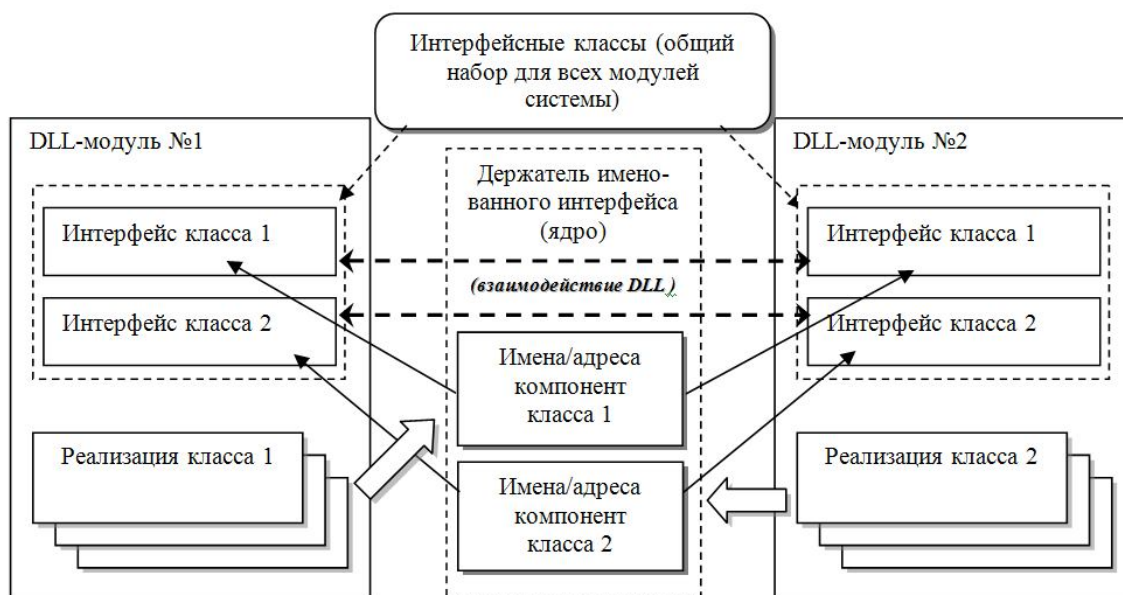


Рисунок 3 – Схема организации межпрограммного взаимодействия посредством интерфейсных классов

Управление интерфейсом обеспечивается с помощью трех базовых функций:

- получения именованного адреса;
- добавления именованного адреса;
- удаления именованного адреса.

Прототипы указанных методов на языке C++ приведены ниже.

```
void *GetNamedAddress(char * AddressName); //Получение адреса
void AddNamedAddress(void * Address,char * AddressName); //Добавление адреса
void DelNamedAddress(char * AddressName); //Удаление адреса
```

Базовые функции реализуют «обеспечение интерфейса» и позволяют получить адрес процедуры или структуры данных (интерфейсного компонента), добавить интерфейсный компонент или удалить интерфейсный компонент с использованием его имени (уникального строкового идентификатора). Методы интерфейса (компоненты), к которым необходимо получить доступ, на языке C++ в общем виде описываются следующим образом:

```
TYPE Fun1(Arg1, Arg2, ..., ArgN)
{
    static TYPE (*fp)(Arg1, Arg2, ..., ArgN)=NULL;
    if (!fp) fp=( TYPE (*fp)(Arg1, Arg2, ..., ArgN)) GetNamedAddress("Name");
    return fp(Arg1, Arg2, ..., ArgN);
}
```

где TYPE – тип возвращаемого значения функции, Arg1, Arg2, ..., ArgN – список аргументов

функции, "Name" – уникальный текстовый идентификатор получаемого интерфейса.

Организованный таким образом компонент интерфейса программно доступен в программе как обычная функция, что не требует никаких дополнительных инициализаций, обеспечивает простоту использования, высокую скорость выполнения (на уровне вызова локальной подпрограммы) и легкость отладки. Все компоненты программного комплекса (DLL-модули) являются равноправными и могут добавлять и использовать интерфейсы (точки входа) в зависимости от поставленной задачи. Добавление и изменение интерфейсных компонентов в программной системе могут производиться динамически в ходе выполнения (путем непосредственного вызова функции «AddNamedAddress» / «GetNamedAddress»), что позволяет создавать интерфейсы с временем жизни выполнения процедуры обработки. Одна и та же интерфейсная функция может иметь разную реализацию (и меняться несколько раз в ходе выполнения программы) в зависимости от выполняемой задачи. Например, вывод отчета может производиться на экран, на принтер или в файл, иметь разную форму, а представляться одним прототипом, например void Report(char *cReportText).

Объединение отдельных компонентов интерфейса в класс позволяет получить интерфейсный класс для взаимодействия с модулем – держателем класса. При расширении функциональности компонента необходимо также расширять и интерфейсный класс. При этом сохра-

нится работоспособность ранее созданных DLL-модулей с использованием предыдущей версии интерфейсного класса, что обеспечивает наследуемость интерфейса.

4. Реализация атрибутивного описания

Атрибутивная информация представляет собой сведения об условиях получения данных ДЗЗ и предыдущих видах обработки, а именно: информация о системе ДЗЗ (тип космического аппарата и датчика); дата, место, время и условия съемки; установочные параметры системы ДЗЗ; радиометрические коэффициенты; параметры выходной проекции и многое другое. В системах обработки ДЗЗ атрибутивная информация является неотъемлемой частью изображений наряду с растровыми данными. В системе «ER-Set» реализованы оригинальные механизмы хранения и обмена атрибутивной информацией (данные хранятся в атрибутивном заголовке, который связан с изображением).

С целью унификации процедур обработки для «ER-Set» разработан внутренний стандарт структуры атрибутивной информации. Вся информация структурирована посредством дерева тэгов. Отдельные параметры в тэгах хранятся в нетипизованном виде – то есть в виде текстовых строк. Тип считываемого параметра формализуется только на этапе его чтения. Атрибутивный заголовок позволяет также хранить массивы данных в бинарном виде. Общий объем атрибутивной информации может составлять сотни мегабайт. Содержательная часть структуры атрибутивной информации в данной статье не приводится, поскольку является достаточно объемной и специфичной, что выводит ее за рамки объема данной статьи.

Техническая реализация доступа к атрибутивному заголовку выглядит на примере языка C++ следующим образом.

```
Image.Header("Тэг\Имя_параметра") = 'Значение';
```

```
«Переменная» = Image.Header("Тэг\Имя_параметра"),
```

где, 'Значение' – переменная или константа любого скалярного типа данных,

Header – класс заголовка изображения (является элементом класса изображения Image).

Приведенная выше запись не типична для языка C++, но является лаконичной и понятной для программистов. Такая синтаксическая конструкция обеспечивает работу с любыми скалярными типами данных и реализуется посредством группы переопределенных операторов, например:

operator &()(const char *cName) – оператор преобразования операнда,

operator float() – оператор чтения вещественного параметра,

operator = (const char *cName) – оператор записи текстового параметра.

Таким образом, в системе «ER-Set» реализован простой (с точки зрения синтаксиса) и унифицированный (с точки зрения различных типов данных) механизм работы с атрибутивным описанием изображений, что позволяет единообразно обрабатывать, а также атрибутивно описывать изображения различных типов – сканерных, матричных, тепловых, радиолокационных и т.п. Единообразие и унификация обеспечивают возможность повторного использования кода (уже разработанных DLL-модулей), что кардинально повышает производительность труда всей организации.

5. Технологии программирования

5.1. Динамическое подключение

В системе «ER-Set» реализовано динамическое (в том числе в ходе выполнения программы) подключение новых DLL-модулей расширения. В качестве модулей расширения могут выступать как DLL-модули импорта/экспорта (появляется возможность импорта/экспорта новых форматов входных данных), так и модули обработки. Подключаемые модули позволяют добавлять новые функции, изменять существующие функции, расширять пользовательский интерфейс (добавлять инструментальные панели, меню, консольные команды) и т.п. Со стороны пользователя программного комплекса динамическое подключение выглядит как открытие стороннего файла (DLL-модуля), содержащего дополнительные функции.

5.2. Виртуальный импорт и обработка

В ядре системы реализовано два механизма обработки данных – процедурный, при котором создается новая копия физической матрицы изображения, и сквозной (виртуальный) механизм, когда обработка данных осуществляется «на лету». При реализации сквозной обработки данных изображения обрабатываются только по необходимости и только в том объеме, который был запрошен. Такой подход наиболее приемлем для обработки сверхбольших объемов данных, поскольку в этом случае отсутствует необходимость в сохранении промежуточных результатов на диске (или в памяти) и ожидании результатов при многостадийном процессе получения выходной продукции.

В наиболее известных зарубежных системах обработки данных дистанционного зондирования ER-DAS и ER-Mapper используется ограниченный набор форматов файлов, которые непосредственно ими импортируются. Для открытия файлов других форматов используются процедуры конвертации, то есть сначала необходимо конвертировать файл в «родной» формат и только затем открыть его. Данный подход является наиболее простым в реализации, но не отвечает требованиям оперативности.

В системе «ER-Set» открытие файлов и преобразование данных во внутренний формат осуществляются модулями-конверторами «на лету», то есть используется механизм виртуального импорта и обработки [7], что обеспечивает большую по сравнению с традиционным подходом оперативность обработки.

Суть метода виртуального импорта и обработки заключается в установлении многостадийного информационного соответствия (в виде цепочки вызовов функций) между данными во внешнем файле и результатом обработки. Такое информационное соответствие позволяет исключить (виртуально) отдельный этап загрузки исходного изображения в память компьютера и его полной обработки. На практике для пользователя это означает, что процесс загрузки и обработки изображения визуально отсутствует, но на самом деле он совмещается с процессом отображения изображения на мониторе или сохранением на диск, что кардинально сокращает

время первичного просмотра и анализа получаемых результатов. То есть обработка производится порционно. Например, порцией данных может быть отображаемый фрагмент изображения на мониторе.

5.3. Технология сборки приложений

Технология сборки готовых приложений основана на использовании универсального ядра программной системы (EXE-модуля) и интеграции его с унифицированными DLL-модулями функционального расширения. Сборка приложений на основе платформы «ER-Set» осуществляется путем формирования каталога программного комплекса в соответствии с заданными правилами. Модули расширения, в соответствии со своим назначением, размещаются в разных поддиректориях.

Типичный программный комплекс, разработанный на основе платформы «ER-Set», состоит из следующих директорий:

kernel, где размещаются модули расширения функций ядра, которые загружаются автоматически при запуске программного комплекса и предоставляют функции, доступные в течение всего сеанса работы приложения;

codec, где размещаются модули импорта/экспорта (также загружаются автоматически);

proc, где размещаются DLL-модули, которые загружаются и выгружаются оперативно во время работы при необходимости.

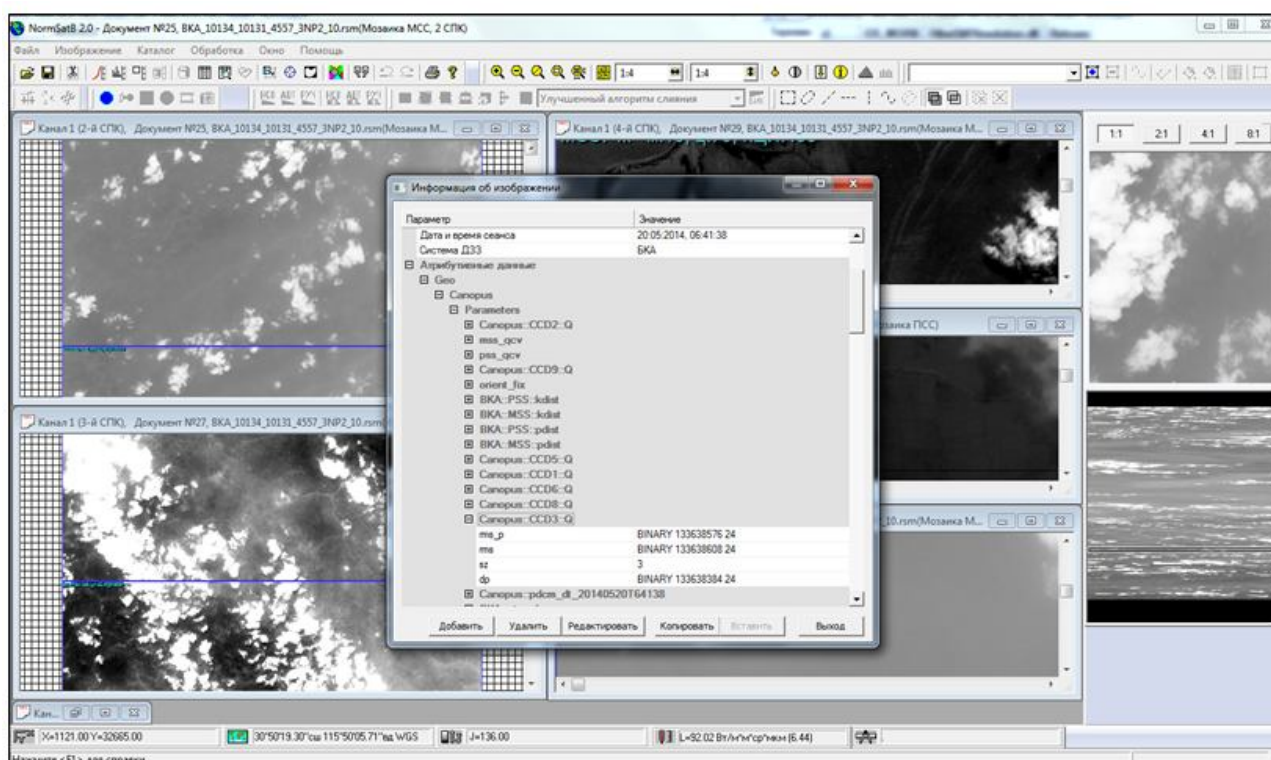


Рисунок 4 – Типичный внешний вид приложения

Дополнительно готовое приложение содержит специальный DLL-модуль пользовательского интерфейса, который определяет основной вид программы (главное меню, название, заставку и т.п.) и содержит графические ресурсы приложения. Модули расширения (расположенные в поддиректории kernel) могут дополнительно расширять пользовательский интерфейс набором инструментальных панелей. На рисунке 4 показан внешний вид типичного приложения, разработанного на основе платформы «ER-Set», на примере программного комплекса «NormSatB2» [2], эксплуатируемого в Научном центре оперативного мониторинга Земли, г. Москва, и предназначенного для обработки данных от систем ДЗЗ «Канопус-В» и «БКА».

6. Технические возможности платформы

Платформа «ER-Set» позволяет реализовывать программные комплексы в операционных средах Windows версий XP, 7, 8, Server с разрядностью 32/64 бит. При реализации 32-разрядного приложения и использования всей оперативной памяти компьютера применяется механизм расширения адресного пространства AWE (Address Windowing Extensions).

Любой программный комплекс, разработанный на основе платформы «ER-Set», обладает техническими характеристиками и возможностями, приведенными ниже.

Ядро «ER-Set» позволяет использовать в работе до 256 Гб оперативной памяти компьютера. Объем обрабатываемых физических данных составляет 1 Тб, виртуальных (динамически формируемых) - не ограничен. Система рассчитана на функционирование на многоядерных и многопроцессорных платформах (в том числе серверных, с несимметричной памятью).

При визуализации предусмотрено отображение растровых данных с разрядностью от 1 до 16 бит, а также растров, представленных в формате с плавающей точкой. На растровые данные могут быть наложены несколько векторных слоев.

В настоящее время DLL-модули в формате «ER-Set» могут быть созданы с использованием языка C++ (в среде Microsoft Visual Studio, начиная с версии 6.0) или на языке Pascal в среде Embarcadero Delphi. Для программистов DLL-модулей разработаны пакет средств разработчика и документация. На текущий момент разработано более 1.5 тысячи DLL-модулей. Для Er-Set разработано множество модулей импорта/экспорта, которые позволяют работать с изображениями в форматах bmp, tiff, jpeg, jpeg2000, img, raw, ascii grid, bil, ers, lan, dem, mrsid, hdf, sxf и других.

В ядре реализована встроенная система отмены (возврата к предыдущему состоянию) произведенных операций с растровыми данными. Реализация данной операция совершенно «незаметна» для программистов и пользователей системы (то есть реализуется полностью автоматически и без заметных затрат времени).

Для программирования пользовательского интерфейса в «ER-Set» предусмотрены перехват и трансляция в DLL-модули событий от средств ввода (клавиатура, мышь, меню, инструментальные панели). Для внутреннего обмена информацией между компонентами пользовательского интерфейса реализован механизм обмена «консольными» сообщениями в виде текстовых строк.

Разработанные программные комплексы на основе платформы «ER-Set» могут защищаться аппаратными ключами на уровне ядра и отдельных модулей.

Заключение

На сегодняшний день платформа «ER-Set» является эффективным и незаменимым инструментом разработки программных комплексов для обработки данных ДЗЗ в НИИ «Фотон». На данной платформе уже реализовано около двух сотен программных комплексов, которые эксплуатируются в России и за рубежом.

Библиографический список

1. **Свидетельство о государственной регистрации программы для ЭВМ № 2013610968.** Универсальная платформа быстрой разработки приложений для обработки данных дистанционного зондирования Земли «ER-Set» / В.И. Побаруев; заявитель и правообладатель ФГБОУ ВПО «Рязанский государственный радиотехнический университет»; заявл. 04.12.2012; опубл. 09.01.2013.
2. **Кузнецов А.Е., Побаруев В.И., Пошихонов В.И.** Программный комплекс обработки информации от сканерно-кадровых съемочных систем КА «Канопус-В» и «БКА» // Современные проблемы дистанционного зондирования Земли из космоса. 2014. № 1. С. 287-300.
3. **Побаруев В.И.** Организация межпрограммного взаимодействия в сложных системах методом «ассоциативного» интерфейса // Вестник Рязанского государственного радиотехнического университета. 2013. № 2(44). С. 56-60.
4. **Побаруев В.И., Светелкин П.Н.** Хэштейловая пирамидальная реорганизация изображений в системах обработки данных дистанционного зондирования Земли // Цифровая обработка сигналов. 2010. № 4. С. 33-37.
5. **Антонушкина С.В., Гуров В.С., Егошкин Н.А., Еремеев В.В., Зенин В.А., Князьков П.А., Козлов Е.П., Кузнецов А.Е., Макаренков А.А., Москвитин А.Э., Побаруев В.И., Пошихонов В.И.,**

Пресняков О.А., Светелкин П.Н. Современные технологии обработки данных дистанционного зондирования Земли: монография под редакцией Еремеева В.В. М.: ФИЗМАТЛИТ, 2015. 460 с.

6. **Эш Рофэйл, Яссер Шохауд.** COM и COM+. Полное руководство: пер. с англ. Киев.: ВЕК+, Киев.:

НТИ, М.: Энтроп, 2000. 560 с.

7. **Побаруев В.И.** Организация эффективного кэширования данных в среде Windows NT 4.0 // Новые информационные технологии в научных исследованиях и в образовании: Всероссийская науч.-техн. конф., тез. докл., Рязань, 2000. С.80-82.

UDC 004.422.81

UNIFIED SOFTWARE PLATFORM FOR CREATING PROCESSING MEANS OF REMOTE SENSING DATA - «ER-SET»

V. I. Pobaruev, Ph.D. (technical sciences), associate professor, Leading Researcher, Photon Research Institute, RSREU, Ryazan, foton@rsreu.ru

The architecture, principles of operation, programming methods and specifications of the software platform developed in "Foton" institute and used to create specialized data processing software for Earth remote sensing systems are considered. The goal is to create a rapid development technology of high-performance applications to process remote sensing data. Software platform is a set of components (templates and modules), binary interfaces and the integration rules of elements into a single system. High performance of software systems created by this platform is provided by technological and architectural solutions. The technology of quick application compilation is based on high unification of developed software (DLL-modules), interfaces, and the rules of integration.

Key words: remote sensing, system software, programming techniques, interface, performance, software architecture, rapid application development, unification.

References

1. **Pobaruev V. I.** Svidetel'stvo o osudarstvennoj registracii programmy dlja JeVM № 2013610968. Universal'naja platforma bystroj razrabotki prilozhenij dlja obrabotki dannyh distancionnogo zondirovanija Zemli «ER-Set»; the applicant and the right holder VPO "Ryazan State Radio Engineering University"; appl. 04.12.2012; publ. 09.01.2013 (in Russian).

2. **Kuznecov A. E., Pobaruev V. I., Poshehonov V. I.** Programmnyj kompleks obrabotki informacii ot skanerno-kadrovых semochnyh sistem KA «Kanopus-V» i «BKA». *Sovremennye problemy distancionnogo zondirovanija Zemli iz kosmosa*. 2014, no 1, pp. 287-300 (in Russian).

3. **Pobaruev V. I.** Organizacija mezhprogrammogo vzaimodejstija v slozhnyh sistemah metodom «associativnogo» interfejsa. *Vestnik Rjazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2013, no 2(44), pp. 56–60 (in Russian).

4. **Pobaruev V. I., Svetelkin P. N.** HESh-Tejlovaja piramidal'naja reorganizacija izobrazhenij v sistemah obrabotki dannyh distancionnogo zondirovanija Zemli.

Cifrovaja obrabotka signalov. 2010, no 4, pp. 33-37 (in Russian).

5. **Antonushkina S. V., Gurov V. S., Egoshkin N. A., Eremeev V. V., Zenin V. A., Knjaz'kov P. A., Kozlov E. P., Kuznecov A. E., Makarenkov A. A., Moskvitin A. Je., Pobaruev V. I., Poshehonov V. I., Presnjakov O. A., Svetelkin P. N.** *Sovremennye tehnologii obrabotki dannyh distancionnogo zondirovanija Zemli: monografija pod redakciej Eremeeva V.V.* (Modern technology processing of Earth remote sensing: a monograph edited by V.V. Eremeev). Moscow.: FIZMATALIT, 2015, 460 p. (in Russian).

6. **Jesh Rofjejl, Jasser Shohaud.** COM i COM+. Polnoe rukovodstvo. Per. s angl. (COM and COM +. Complete Guide: Per. from English.). Kiev.: VEK+, Kiev.: NTI, Moscow.: Jentrop, 2000, 560 p. (in Russian).

7. **Pobaruev V. I.** Organizacija jeffektivnogo kjeshirovanija dannyh v srede Windows NT 4.0. *Novye informacionnye tehnologii v nauchnyh issledovanijah i v obrazovanii: Vserossijskaja nauch.-tehn. konf., tez. dokl., Rjazan', 2000*. pp.80-82 (in Russian).