

УПРАВЛЕНИЕ СОПРОВОЖДЕНИЕМ СЛОЖНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ С ИЕРАРХИЧЕСКОЙ СТРУКТУРОЙ

Орлов В.Л., Курако Е.А.

Институт проблем управления им. В.А. Трапезникова РАН
ovl@ipu.ru, kea@ipu.ru

Аннотация: Рассмотрены особенности управления сопровождением сложных информационных систем с иерархической структурой. Выделена задача обновления программ и данных, как одна из основных в процессе сопровождения. Предложен сценарный подход к организации обновления в распределенных системах. Определены основные методы обновления программного обеспечения и информационных структур серверов приложений, серверов баз данных и клиентских мест.

Ключевые слова: сопровождение; обновление программ; сложные информационные системы; сервер приложений; база данных; клиент.

Введение

Задача управления сопровождением сложных информационных систем в последнее время приобретает все большую актуальность.

Сопровождение в общем случае – это объемная работа, включающая:

- контроль за работой оборудования и восстановление;
- контроль функционирования операционных систем (ОС) и восстановление;
- контроль сетевого обеспечения и восстановление;
- контроль прикладного программного обеспечения (ПО) и восстановление;
- обновление прикладного ПО при выявлении ошибок в функционировании или в случае введения новых возможностей, когда разработчики представили новую версию.

При этом, если контроль и восстановление функциональных возможностей оборудования, операционных систем, сетевого обеспечения и даже прикладного ПО могут проводиться локально, то обновление прикладных программ – это, как правило, комплексная задача, требующая проведение синхронной модификации всей системы в целом. Сложность задачи определяется следующими моментами:

- сложностью архитектуры систем, которые содержат несколько уровней, причем уровни могут быть территориально распределены;
- разнородностью внутриуровневого построения, которое предусматривает как минимум три принципиально различных базовых компонента: клиентское место, сервер приложений и сервер баз данных (БД);
- необходимостью синхронизации изменений не только внутри уровня, но и между уровнями;
- возможностью только кратковременных остановок, предназначенных для изменения программного обеспечения, или (что желательно) отсутствия таковых.

Естественно, что наличие всех этих условий делает задачу нетривиальной. В то же время ее решение необходимо, так как практика использования программ любой сложности показывает, что их модернизация в процессе жизненного цикла проводится всегда. Это позволяет не только устранять ошибки, но и расширять функциональные возможности.

Таким образом, обновление прикладного ПО является необходимой частью процесса сопровождения. При обновлении требуется принимать во внимание различие типов обновляемой информации, особенность построения фрагментов, составляющих систему, их взаимосвязи и принципы синхронизации.

В связи с этим далее рассматриваются методы обновления программ и данных в сложных распределенных информационных системах с иерархической структурой.

1 Сценарный подход в процессе подготовки коррекции программ и данных

Как отмечало выше, на каждом уровне системы могут быть выделены такие компоненты, как клиентские места, серверы приложений и серверы БД (рису 1).

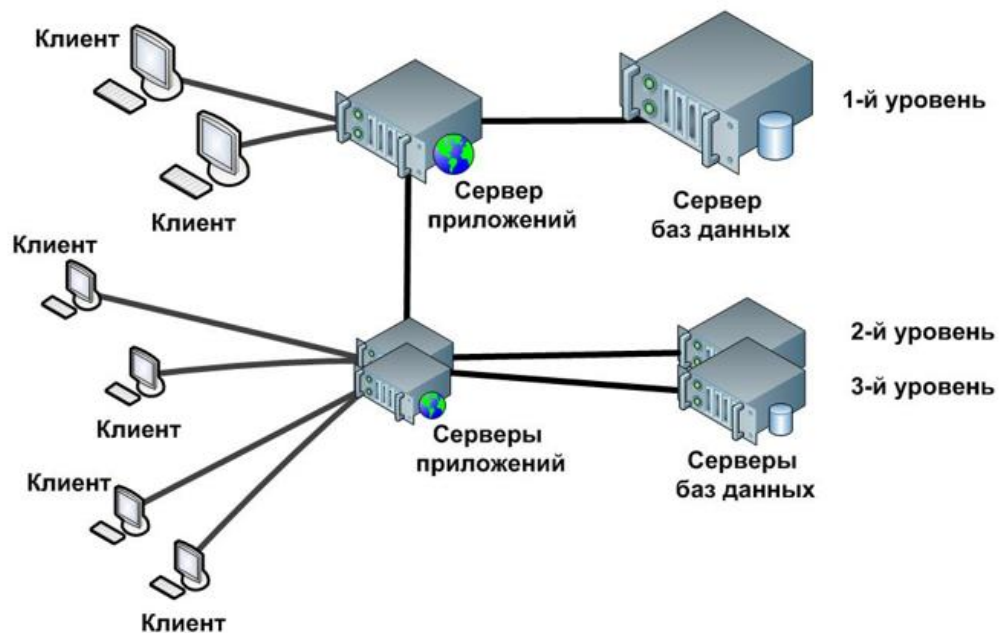


Рис. 1. Уровни системы и компоненты одного уровня

Из всего многообразия объектов, представленных на компонентах, целесообразно выделить следующие [1].

- на клиенте:
 - программные модули и библиотеки;
 - директории и файлы;
- на сервере приложений:
 - программные модули и библиотеки;
 - директории и файлы;
 - службы операционной системы;
 - электронные сервисы;
- на сервере баз данных:

- хранимые процедуры;
- структура БД (таблицы, представления, последовательности, связи и т.д.).

Размещение объектов на компьютерах, принадлежащих одной ветке иерархии, иллюстрирует рис.2.

Необходимо отметить, что можно выделить разные типы клиентов: «толстый», «тонкий» и «сервис-браузер» [2,3]. Сопровождение «толстого» клиента проводится обычно автономно. «Тонкий» клиент не включает в свой состав программные модули, библиотеки, директории и файлы, которые периодически обновляются со стороны сервера. «Сервис-браузер» представляет собой более общий случай, который подразумевает циклическое обновление перечисленных выше объектов. Поэтому при дальнейшем рассмотрении управления сопровождением систем, в основном, акцент делается на системы, использующие сервис-браузерную технологию.

При использовании сервис-браузерной технологии на клиенте первоначально размещается только сервис-браузер. В зависимости от прикладной системы, которую сервис-браузер обслуживает, он обращается к серверу приложений, чтобы скопировать на клиентский компьютер прикладные модули, программные библиотеки, файлы и директории, содержащие данные (рис.2). То есть сервис-браузерная технология изначально подразумевает включение механизма обновления. Причем обновление проходит даже на начальном этапе, когда модули, библиотеки, файлы и директории просто отсутствуют на компьютере. На этом этапе процесс обновления трансформируется в начальную установку.

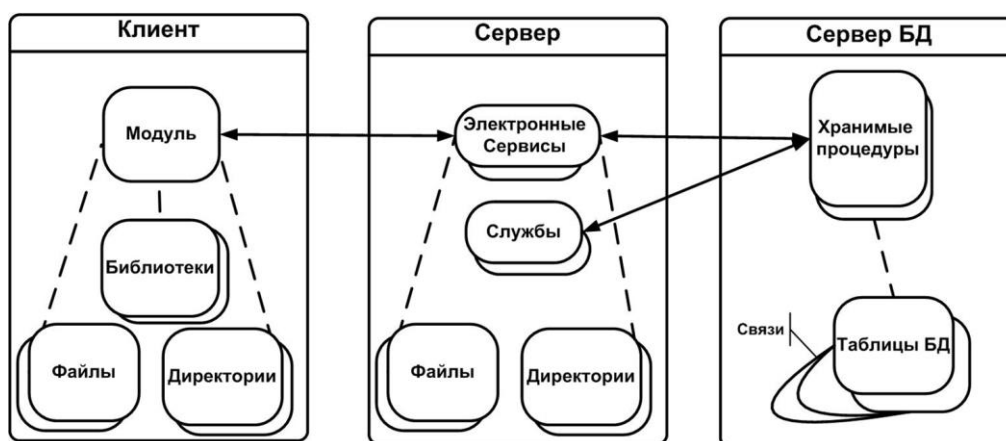


Рис. 2. Размещение объектов на компьютерах разного типа

Обработка объектов разных типов ведется различными способами. Основные типы обработки представлены в таблице 1.

Таблица 1. Виды обработки объектов

Объекты	Виды обработки
Директории	Создание, копирование, удаление
Файлы	Копирование, удаление
Модули	Копирование, удаление
Библиотеки	Копирование, удаление
Службы и сервисы	Регистрация, копирование, удаление
Хранимые процедуры БД	Создание, замена, удаление
Структура БД	Создание, модификация, удаление
Управляющая информация системы, хранящаяся в БД	Создание, модификация, удаление

Как мы видим, обработка различных типов объектов проводится по-разному. В то же время целесообразно проводить группировку обработки нескольких объектов или даже нескольких типов объектов в одном пакете. Это объясняется тем, что, например, при добавлении какой-либо новой функции требуется изменение как программ клиента, так и серверных программ, а также необходимо проведение коррекции БД. Все эти действия в необходимой последовательности могут быть корректно выполнены, если их реализация будет определяться сценарием.

Поэтому вместе с подготовкой пакета коррекции необходимо для каждого объекта указать его имя, тип объекта, а также место хранения информации и место, где ее требуется разместить. Все эти данные попадают в автоматически формируемый сценарий.

То есть в итоге пакет включает в свой состав сценарий, в каждой строке которого хранится описание объекта и действий, которые с ним нужно совершить при установке, а также все необходимые объекты, указанные в сценарии (рис.3).

При этом важно иметь виду следующее обстоятельство: тело объекта может быть двух видов. Если это информационный блок или программа, то тело пакета представляет собой файл определенного формата. Замещение тел объектов этого вида проводится копированием. Другой вид представляет собой сценарий (скрипт), при исполнении которого создается, изменяется или удаляется фрагмент системы. Например, необходимо добавить новую таблицу в БД, то есть изменить структуру данных. В данном случае изменение БД производится путем выполнения скрипта. Здесь важно отметить, что при наличии таких объектов в пакете появляются сценарии второго уровня.

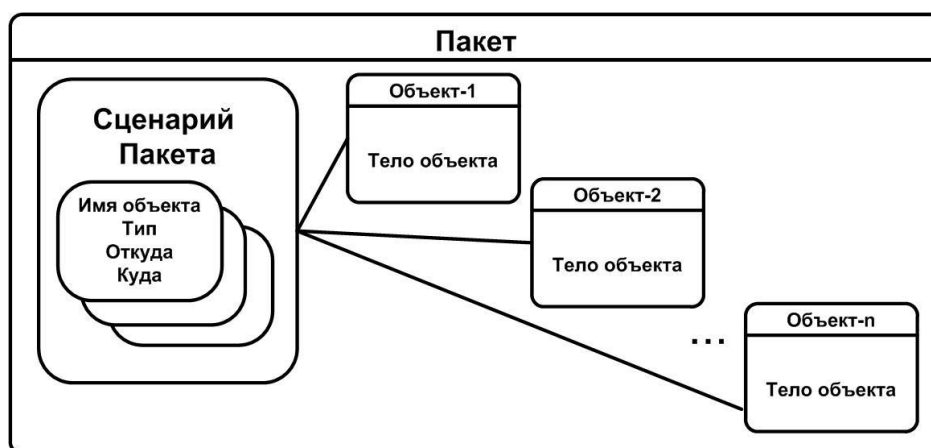


Рис. 3. Структура пакета обновлений

2 Формирование сценария разработчиком

На первом этапе проводится разработка объектов с использованием языков программирования. При этом программные модули клиента, относящиеся к ним библиотеки и объекты, содержащие данные, необходимые для функционирования модулей и библиотек, оформляются в виде файлов. Совокупность файлов с данными может быть представлена в виде директории. Измененные электронные сервисы и службы операционной системы обычно также формируются в виде директорий.

Все перечисленные выше файлы и директории могут быть отнесены к объектам первого типа.

Хранимые процедуры и изменения в структуре базы данных обычно представляют собой файлы, содержащие скрипты, которые должны быть выполнены в процессе обновления. В дальнейшем эти файлы будем называть объектами второго типа.

Когда все объекты первого и второго типа подготовлены, требуется выполнить два действия:

1. разместить все объекты первого типа в соответствующих местах информационной системы, заменив или дополнив существующий набор объектов;
2. переместить все объекты второго типа в те фрагменты информационной системы, где они могут быть выполнены, проведя необходимую коррекцию программ и данных.

То есть любое действие А может быть составлено из трех элементов:

- перемещение М;
- копирование С;
- выполнение Е.

Если для объектов первого типа используют перемещение и копирование, то для объектов второго типа – перемещение и исполнение. То есть

$$A_1 = M + C$$

$$A_2 = M + E$$

Перемещение может быть организовано несколькими способами. Первый способ – рассылка информации по адресам. Это вызывает определенные трудности, так как центральный сервер, должен знать адреса всех компьютеров, где требуется установка того или иного объекта. При проявлении новых компьютеров в сети необходимо корректировать список рассылки. Второй способ – запрос информации с сервера, получающего информацию обновления с одного из серверов предыдущего уровня. В этом случае проблема масштабирования решается просто, так как компьютер определенного типа, подключенный к сети в произвольной точке, сам начинает запрашивать информацию обновления. Для этого ему достаточно иметь один адрес. Возможен вариант, когда в сети имеется один центральный сервер обновления и он может выдать информацию в зависимости от вида запроса при условии предъявления соответствующих полномочий.

Сценарий можно формировать автоматически по указаниям администратора. Для этого может использоваться программа формирования сценариев [3]. На рис.4 приведено основное окно этой программы.

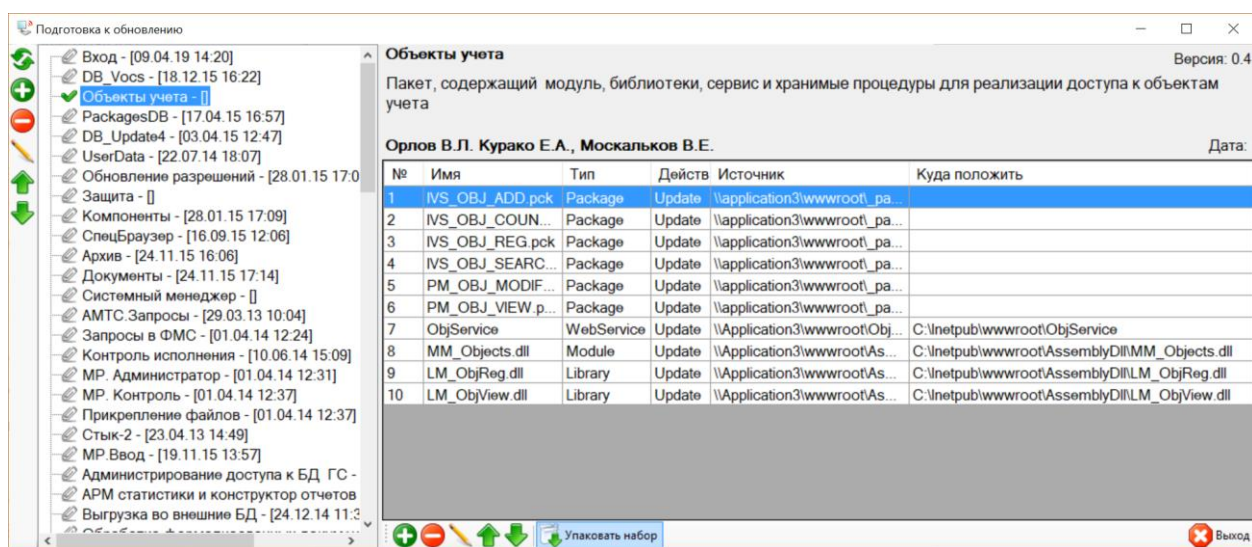


Рис. 4. Основное окно программы формирования сценариев

В ситуации, отображенной на рис.4. проводится обработка пакета, который называется «Объекты учета». В правой части отображается состав пакета. Он содержит модуль (Module) MM_Object.dll, библиотеки (Library) LM_ObjReg.dll, LM_ObjView.dll, web-сервис (WebService) ObjService и хранимые процедуры для БД (Package) IVS_OBJ_ADD.pck и другие.

В результате работы программы должно быть сформировано два файла:

- 1) упакованный набор тел объектов;
- 2) сценарий пакета.

Для формирования файлов необходимо нажать кнопку «Упаковать набор». При этом все тела объектов помещаются в один архивированный файл. Этот файл будет выбираться серверами приложений в процессе периодических запроса на обновление, распаковываться и размещаться (или выполняться) в соответствии со сценарием пакета.

Автоматически сформированный сценарий пакета в формате XML приведен ниже

```
<?xml version="1.0" encoding="utf-8" ?>
<Instructions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<InstructionSet>
<Instruction>
<DestPath />
<SourcePath>\\application3\wwwroot\_packages\IVS_OBJ_ADD.pck</SourcePath>
<Action>Update</Action>
<Type>Package</Type>
</Instruction>
<Instruction>
<DestPath />
<SourcePath>\\application3\wwwroot\_packages\IVS_OBJ_COUNTRY.pck</SourcePath>
```

```

<Action>Update</Action>
<Type>Package</Type>
</Instruction>
<Instruction>
<DestPath />
<SourcePath>\\application3\wwwroot\_packages\IVS_OBJ_REG.pck</SourcePath>
<Action>Update</Action>
<Type>Package</Type>
</Instruction>
<Instruction>
<DestPath />
<SourcePath>\\application3\wwwroot\_packages\IVS_OBJ_SEARCH.pck</SourcePath>
<Action>Update</Action>
<Type>Package</Type>
</Instruction>
<Instruction>
<DestPath />
<SourcePath>\\application3\wwwroot\_packages\PM_OBJ_MODIFY.pck</SourcePath>
<Action>Update</Action>
<Type>Package</Type>
</Instruction>
<Instruction>
<DestPath />
<SourcePath>\\application3\wwwroot\_packages\PM_OBJ_VIEW.pck</SourcePath>
<Action>Update</Action>
<Type>Package</Type>
</Instruction>
<Instruction>
<DestPath>C:\Inetpub\wwwroot\ObjService</DestPath>
<SourcePath>\\Application3\wwwroot\ObjService</SourcePath>
<Action>Update</Action>
<Type>WebService</Type>
</Instruction>
<Instruction>
<DestPath>C:\Inetpub\wwwroot\AssemblyDll\MM_Objects.dll</DestPath>
<SourcePath>\\Application3\wwwroot\AssemblyDll\MM_Objects.dll</SourcePath>
<Action>Update</Action>
<Type>Module</Type>
</Instruction>
<Instruction>
<DestPath>C:\Inetpub\wwwroot\AssemblyDll\LM_ObjReg.dll</DestPath>
<SourcePath>\\Application3\wwwroot\AssemblyDll\LM_ObjReg.dll</SourcePath>
<Action>Update</Action>
<Type>Library</Type>
</Instruction>
<Instruction>
<DestPath>C:\Inetpub\wwwroot\AssemblyDll\LM_ObjView.dll</DestPath>
<SourcePath>\\Application3\wwwroot\AssemblyDll\LM_ObjView.dll</SourcePath>
<Action>Update</Action>
<Type>Library</Type>
</Instruction>
</InstructionSet>
<Comment>Пакет, содержащий модуль, библиотеки, сервис и хранимые процедуры для
реализации доступа к объектам учета</Comment>
<Name>Объекты учета</Name>
<Version>0.4</Version>
</Instructions>

```

В сценарии указываются все объекты, входящие в пакет, их типы, определяющие действия, а также исходное месторасположение объектов и место назначения на сервере приложений.

3 Алгоритм обработки сценария на серверах

Важным элементом сценария является номер версии пакета. Он используется при проверке необходимости обновления. Все подготовленные пакеты помещаются на центральный сервер обновлений (ЦСО). В сценарии каждого пакета содержится номер его версии.

Все периферийные службы обновления (ПСО) периодически обращаются к ЦСО, который предоставляет им список хранящихся у него пакетов, содержащих обновления. В списке у каждого пакета присутствует номер его версии.

После получения списка ПСО сверяет список с имеющимся у нее списком установленных обновлений, выбирает первое неустановленное обновление, запрашивает его у ЦСО и производит установку. Здесь важным является то обстоятельство, что список задает последовательность установки обновлений. Это существенно, так как нарушение этой последовательности может привести к невозможности инсталляции того или иного пакета. То есть здесь работает метод исключения пропуска установки. Пока не установится определенное обновление, последующие также не будут устанавливаться.

Например, если подготовленный пакет по каким-либо причинам содержит ошибку, которая не дает возможности его установить, то запросы от ПСО будут периодически повторяться, до тех пор, пока в ЦСО не появится новая корректная версия пакета.

Если же вновь установленный пакет не удовлетворяет сотрудников, обеспечивающих эксплуатацию, то возможен возврат на предыдущую версию по команде оператора ПСО. В этом случае для обновления будет приниматься только следующая версия, номер которой превышает номер версии, забракованной на этом сервере.

4 Обновление сервера приложений

На сервере приложений обновляются файлы, библиотеки, директории, электронные сервисы и службы, связанные с операционной системой. Если файлы замещаются простым копированием в соответствии с параметрами, указанными в сценарии, то электронные сервисы и службы, необходимо, как правило, регистрировать. При модификации службы или электронного сервиса требуется сначала их отключить, дождаться завершения работы, заменить необходимые файлы, а затем заново запустить. Эти дополнительные действия определяются типом объекта, также указанным в сценарии. Отметим, что при обработке объекта «директория» иногда необходимо создавать пустые директории. В таком случае используется не процедура копирования, а команда создания директории.

5 Обновление базы данных

База данных не обновляется копированием. Здесь для изменения структуры базы данных, что часто является необходимым действием при сопровождении, и для изменения хранимых процедур используются SQL-скрипты, выполняемые службой обновления на сервере приложений в соответствии со сценарием и корректирующие связанную базу данных. То есть при обновлении баз данных фактически используются вложенные сценарии.

Важным обстоятельством является то, что при подготовке новых модулей и библиотек для клиента их необходимо зарегистрировать во фрагменте БД, содержащим управляющую информацию системы. Такая регистрация проводится службой обновления на сервере приложений.

6 Обновление клиента

Программное обеспечение клиента обладает двумя особенностями. Во-первых, клиентские программы взаимодействуют с БД только через сервисы. Следовательно, они ничего не знают о структуре БД и принципиально не могут выполнять никакие SQL-скрипты. Во-вторых, программное обеспечение клиента не включает в свой состав электронные сервисы и службы. То есть здесь не требуется проводить процедуры регистрации, связанные с операционной системой. Из таблицы 1 следует, что в этом случае для обновления программного обеспечения достаточно использовать простые процедуры создания, копирования и удаления.

Таким образом, для выполнения функций обновления клиента служба обновления размещает в заданном каталоге сервера приложений обновляемые модули, библиотеки, файлы и директории

клиента (обобщенные файлы). Управляющая программа клиента в определенные моменты времени, например, при перезапуске прикладной программы клиента производит сравнение имеющихся в каталоге обобщенных файлов на сервере с аналогичными файлами клиента. Все модули, библиотеки, файлы и директории, которые были обновлены на сервере, копируются в соответствующие каталоги клиента и автоматически подхватываются при старте.

Заключение

Одним из важных элементов процесса сопровождения сложных программных комплексов является обновление программ, структуры данных и данных на всех уровнях. В связи с тем, что в этом случае на всех уровнях используются разнообразные процедуры, определяемые типом объектов обновления, целесообразно применение сценарного подхода. Применение вложенных сценариев, обеспечивает выполнение процедур обновления для всех основных типов объектов. Вложенные сценарии задают два основных типа обработки: производится либо копирование объектов (по умолчанию), либо выполнение сценариев второго уровня, помещенных в тело того или иного объекта. Автоматизация формирования сценариев обеспечивается программным средством АОРИС, который также предоставляет службы и web-сервисы для реализации процедуры обновления согласно сценариям.

Литература

1. *Курако Е.А., Орлов В. Л.* Методы динамического обновления программных комплексов для систем с каскадной структурой // Проблемы управления. М., 2017. № 5. С. 52-60.
2. *Курако Е.А., Орлов В. Л.* Сервис-браузеры для информационных систем // Программная инженерия. М., 2017. Т. 8, № 9. С. 413-421.
3. *Kurako E.A. Orlov V.L.* Service-browser Architecture and Large-scale Information Systems / Proceedings of the 11th International Conference "Management of Large-Scale System Development" (MLSD). Moscow: IEEE, 2018. С. <https://ieeexplore.ieee.org/document/8551831>.
4. *Курако Е.А.* Система автоматического обновления программного обеспечения в распределенных информационных системах (АОРИС). Правообладатель: ИПУ РАН. Свидетельство о государственной регистрации программы для ЭВМ № 2012619236 от 23.08.2012.