

## Реализация событийно управляемой архитектуры быстрого коллективного доступа к информационным ресурсам ДЗЗ на базе технологии Stream Handler

Д.М. Ермаков<sup>1,2</sup>, К.С. Емельянов<sup>3</sup>, В.П. Саворский<sup>1</sup>, А.П. Чернушич<sup>1</sup>

<sup>1</sup> *Институт радиотехники и электроники РАН, Фрязинский филиал  
Фрязино, Россия*

*E-mail: dima@ire.rssi.ru*

<sup>2</sup> *Институт космических исследований РАН, Москва, Россия*

<sup>3</sup> *Научный центр оперативного мониторинга Земли, Москва, Россия*

Создание информационных систем, обеспечивающих быструю поставку актуальной информации о зонах стихийных бедствий в удобном для потенциальных потребителей виде, является одной из наиболее востребованных задач спутникового мониторинга Земли. Представлены результаты программного макетирования такого рода системы на базе технологии Stream Handler. Важными преимуществами Stream Handler являются высокий уровень абстрагирования внедряемого решения в виде блок-схемы и возможность модификации (по запросу пользователя) и самомодификации (как отклика на некоторые «запускающие» события) алгоритма обработки непосредственно в процессе исполнения. В качестве запускающих использованы сообщения о сейсмических событиях. Макет реализован в виде интегрированного сервиса ЦОХКИ ИРЭ РАН, ориентированного на использование информационных ресурсов типовых российских центров космических данных.

**Ключевые слова:** распределенные информационные системы, событийно управляемая архитектура, Stream Handler.

### Введение

Создание информационных систем, обеспечивающих быструю поставку актуальной информации о зонах стихийных бедствий в удобном для потенциальных потребителей виде, является одной из наиболее востребованных задач спутникового мониторинга Земли. Такие системы полезно рассматривать как важную составную часть распределенных структур коллективного доступа к информационным ресурсам дистанционного зондирования Земли (ДЗЗ), в которых существенную роль играют надежность и эффективность (минимизация времени доставки) обеспечения максимального числа пользователей актуальными информационными продуктами, генерируемыми и / или поставляемыми по запросам, получаемым и обрабатываемым в режиме реального времени.

Один из возможных подходов к созданию подобных информационных систем состоит в реализации ранее предложенной и математически обоснованной идеи событийно управляемой архитектуры (Savorskiy, Tishchenko, 2008; 2009). В рамках данного подхода для оптимизации скорости доступа к информационным продуктам вводится понятие запускающего события. Запускающее событие может состоять, в частности, в возникновении / развитии техногенной катастрофы или стихийного бедствия. В этом случае, в зависимости от типа события, оказывается возможным спрогнозировать изменение актуальности (и, соответственно, ожидаемой частоты запросов) различных информационных продуктов и упреждающим образом реорганизовать структуру распределенных информационных

систем данных ДЗЗ (ИС ДЗЗ) так, чтобы оптимизировать скорость обеспечения информацией всех потенциальных пользователей. В более широком смысле запускающим событием может стать изменение оценки актуальности информационных продуктов на основе собственного анализа статистики запросов, выполняемого системой в фоновом режиме. Запускающим событием может стать и запрос на генерацию некоторого уникального информационного продукта; в последнем случае новый продукт может быть в принципе включен в номенклатуру стандартного набора.

Следует отметить, что преимуществом указанного подхода является возможность его реализации в уже созданных ИС ДЗЗ без кардинальной перестройки их инфраструктуры. Вместо этого необходимо дооснастить существующую ИС ДЗЗ дополнительными средствами ранжирования информационных продуктов в соответствии со степенью их текущей востребованности и упреждающего перемещения их на носители, обеспечивающие оптимальную скорость доступа к продуктам в зависимости от их ранга (чем выше ранг, тем выше скорость доступа или меньше время доставки). В настоящей работе проанализированы возможности создания подобных программных средств, работающих в автоматическом режиме с использованием специализированной технологии обработки потоковых данных, и рассмотрен прототип программного макета с применением этой технологии для конкретного случая, когда запускающими событиями являются глобально регистрируемые землетрясения. Материал изложен следующим образом. В первом разделе дано краткое описание использованной технологии Stream Handler (<http://www.streamhandler.com/>) и основных принципов разработки программного обеспечения на основе этой технологии. Во втором разделе обсуждены возможности применения данной технологии для реализации событийно-управляемой архитектуры в ИС ДЗЗ. В третьем разделе приведен пример разработанного программного макета.

### **Программная архитектура Stream Handler**

Stream Handler – программная архитектура для разработки приложений распределенной обработки потоковых данных. В ее основу положено представление алгоритма в виде графа (блок-схемы) с возможностью независимого исполнения по нескольким параллельным ветвям. Конструктивными элементами системы являются блоки с входами и выходами («контактами») и соединения («провода»).

Блоки абстрагируют понятие операции над данными. По общему назначению их можно разделить на три категории. Блоки чтения («источники») начинают жизненный цикл данных в исполняемой блок-схеме: осуществляют загрузку данных из файлов или получение их от периферийных устройств (в том числе по сети) или из буферов обмена с другими приложениями. Блоки обработки («фильтры») осуществляют различные операции преобразования данных: фильтрацию, поиск, сортировку, объединение и т.п. Блоки отображения («мониторы») заканчивают жизненный цикл обработанной порции данных в исполняемой блок-схеме: выполняют визуализацию данных, запись их в файлы, передачу через буферы обмена другим приложениям, вывод на периферийные устройства.

Реализованные в блок-схеме операции, как правило, имеют параметрическое описание, каждый блок позволяет индивидуальную настройку путем модификации параметров.

Способ внутреннего представления данных во многом определяет адекватность и эффективность применения той или иной программной архитектуры при решении практической задачи. В системе Stream Handler описано несколько базовых типов данных, а также предусмотрена возможность определения новых типов данных и конструирования различных составных структур («контейнеров»), аналогичных структурам языка С. К базовым типам Stream Handler относятся: видеоданные (кадры потока видео или отдельные растровые изображения), текстовые строки, прямоугольники и другие геометрические фигуры, числа (целые и дробные) и массивы чисел, а также некоторые другие (логические значения, метафайлы, аудиоданные, синхроимпульсы).

Для эффективного обмена данными между блоками в процессе исполнения блок-схемы в системе Stream Handler введено абстрактное представление порции данных – пакет. Каждый пакет характеризуется параметрами, необходимыми для синхронизации потоков: уникальным (внутри своего потока) идентификатором и абсолютным временем создания, а также целочисленным идентификатором типа содержащихся в нем данных и некоторыми другими (например, признаком окончания потока, счетчиком обращений). Одним из специфических типов пакетов является контейнер, который содержит в качестве данных ссылки на другие, «вложенные в него» пакеты. С помощью контейнеров можно моделировать древовидные структуры данных любой глубины вложения. Контейнеры являются важным элементом синхронизации данных, т.к. логически объединяют любое число пакетов в совокупность с общим идентификатором и временем создания.

Для повышения удобства и эффективности программного моделирования предметной области предусмотрена возможность определения новых типов данных. Новые типы данных описываются разработчиком на высокоуровневом объектном языке программирования и интегрируются в систему через динамически присоединяемые библиотеки. После этого они используются в системе наравне с базовыми типами данных.

Совокупность использованных в блок-схеме блоков обработки (с их индивидуальными настройками) и установленных между ними соединений полностью описывает логику работы приложения распределенной обработки потоковых данных. Для ее формального описания в системе Stream Handler предусмотрен язык сценария, состоящий из команд модификации блок-схемы. Сценарное описание блок-схемы может быть сохранено в виде специального файла («скрипта») и использовано для дальнейших многократных запусков блок-схемы или ее внедрения в качестве одного из компонентов в более сложную распределенную систему. Для создания приложения на базе системы Stream Handler в принципе достаточно написать соответствующий скрипт, подключить к разрабатываемому приложению динамическую библиотеку ядра Stream Handler и при запуске приложения передать написанный скрипт на исполнение ядру Stream Handler.

Однако процесс разработки, отладки и тестирования блок-схемы удобно проводить в специальной интегрированной среде разработки – Stream Handler Studio, предоставляющей интуитивно ясные средства визуального программирования. В этом случае построение

блок-схемы заключается в ее сборке из отдельных выбираемых пользователем блоков и установлении соединений между ними. Важной особенностью системы Stream Handler является возможность редактирования и настройки блок-схемы в процессе ее исполнения. Это особенность может быть использована как в процессе отладки блок-схемы, так и для создания гибких сценариев, перестраивающих алгоритм работы в зависимости от поступающей информации.

Встроенные библиотеки системы Stream Handler содержат коллекции блоков обработки, реализующих наиболее общие и часто используемые операции работы с данными. Практическая разработка часто требует применения специфических операций, не включенных в стандартные наборы. Поэтому система Stream Handler предоставляет разработчику инструменты проектирования и внедрения собственных блоков обработки данных. Для облегчения этой процедуры предусмотрены мастера создания блоков обработки и библиотеки блоков, интегрированные со средами разработки Microsoft Visual Studio 2008 и Microsoft Visual Studio 2010.

Библиотеки блоков обработки Stream Handler являются одновременно стандартом хранения и переноса (на другие машины с ОС семейства Windows) бинарного кода исполняемых блоков в системе Stream Handler и инструментом включения в систему новых блоков, программируемых разработчиком. Для создания новой библиотеки разработчику требуется только указать директорию размещения и имя проектируемой библиотеки.

Аналогичным образом в созданной библиотеке можно с помощью мастера создания блоков обработки проектировать и реализовывать новые блоки. Разработчик определяет категорию проектируемого блока и назначает ему новое имя. Мастер создает шаблон программного кода для нового блока, который разработчик должен расширить путем переопределения собственно процедуры обработки данных и, при необходимости, описания настроечных параметров блока. Откомпилированная библиотека вместе со всеми блоками становится новым подключаемым модулем (plug-in) системы Stream Handler. Стандарт Stream Handler позволяет также описание новых типов данных для использования в блоках обработки.

### **Событийно-управляемая архитектура ИС ДЗЗ на базе Stream Handler**

Событийно-управляемая архитектура быстрого коллективного доступа к информационным ресурсам ДЗЗ должна обеспечивать реализацию следующих основных функций: опрос специализированных серверов; мониторинг активности пользователей; периодическое обновление ранга важности ресурсов ДЗЗ; генерацию предоставляемых пользователю просмотровых продуктов. Актуальные для рассматриваемой задачи типы данных можно разделить на категории: запросы, мониторинг, собственно данные ДЗЗ и просмотровые продукты. К категориям запросов и мониторинга отнесены различные типа распространяющихся в системе сообщений: запросы исходят от пользователей системы и обрабатываются в асинхронном режиме, мониторинг – это периодически генерируемые системой команды для получения информации о новых событиях. Обе эти категории моделируются одним типом данных, поэтому далее они объединены в группу «сообщений». Все виды сообщений

могут быть промоделированы стандартным типом данных Stream Handler, предназначенным для обмена текстовыми строками (CMainFrame). При этом существует готовый функционал (набор блоков) для основных операций с сообщениями: генерации, форматирования, поиска и выделения фрагментов по шаблону, визуализации на экране. Имеются и более специфические блоки работы с текстовыми сообщениями, например, интерпретатор GPS-сообщений, консольный командный процессор для ОС семейства Windows и т.д.

К одному из наиболее востребованных пользователями виду просмотревых продуктов следует отнести просмотревые изображения: статическое (кадр/сеанс съемки), динамическое (анимированное), мозаику изображений, а также многослойные изображения, включающие дополнительные слои геометрических фигур, подписей и т.п.

Статическое растровое изображение соответствует стандартному типу данных Stream Handler (CVideoFrame). Другие типы изображений моделируются в Stream Handler последовательностью нескольких пакетов данных либо контейнером – пакетом, объединяющим разные типы данных в общую структуру. Для обработки растровых изображений в системе Stream Handler развит мощный набор блоков обработки, включающий различные операции линейной и нелинейной фильтрации, поиска и выделения объектов, геометрических трансформаций, сегментации, цветовых преобразований. Предусмотрен блок отображения растровых изображений, включающий возможность совместного показа нескольких логических слоев (растр, векторная графика, текстовые подписи). Применяются и другие способы визуализации: двумерные графики (профили, разрезы, сечения), трехмерные графики (рельефы поверхности, пространственные спектры), форматированный текст (метаданные, матрицы, координаты и время, единицы измерения). Для построения двумерных графиков и границ двумерных областей в Stream Handler используется стандартный тип данных CFigureFrame. Имеется набор стандартных блоков обработки и преобразования данных этого типа, а также блок визуализации PlotGraph. Рельефы и другие трехмерные графики могут быть визуализированы с помощью блока Render3DPlot. Форматированному тексту соответствует тип CMainFrame, обсужденный выше.

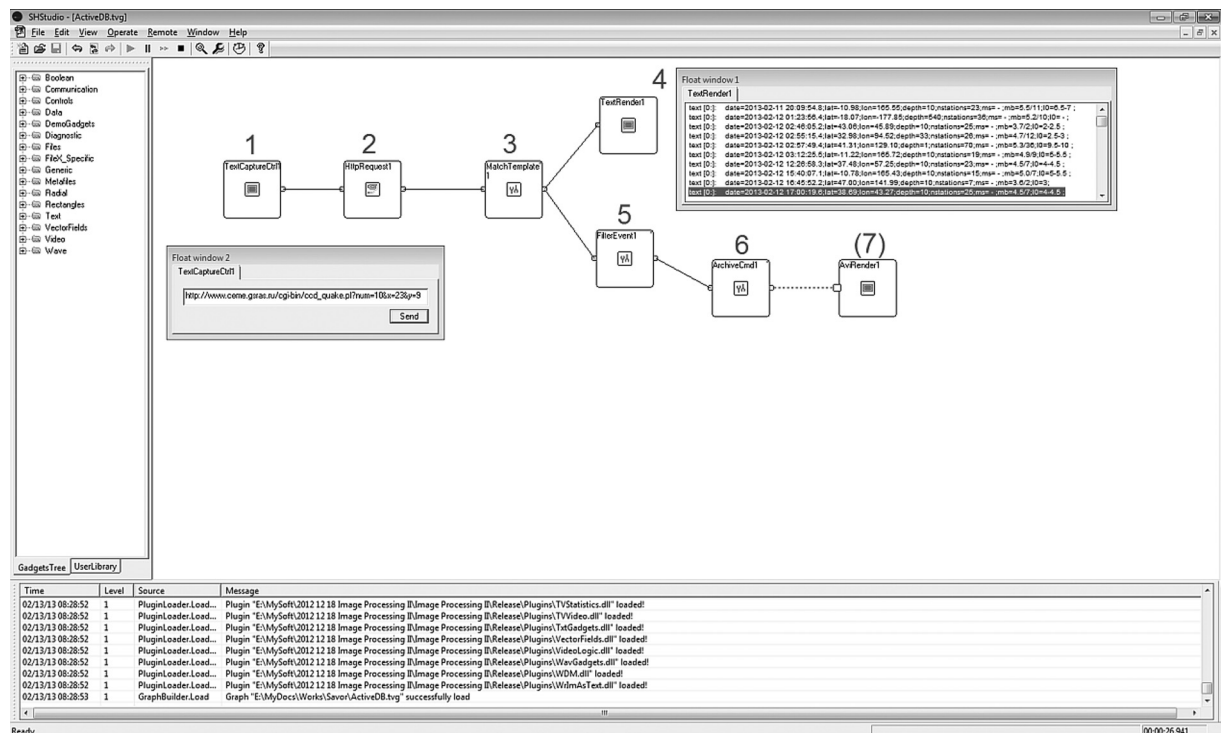
Stream Handler позволяет проводить интеграцию разнородных источников данных в реальном времени, семантически связывать объекты (по положению, времени или иному событийному маркеру) и представлять их в интуитивно понятной форме в интерактивном процессе просмотра выходных данных.

Категория собственно данных ДЗЗ не имеет адекватного аналога в стандарте данных Stream Handler и подлежит в дальнейшем широкому программному моделированию. Однако поставленная на данном этапе узкая задача разработки программного макета позволила свести это моделирование к минимуму.

### **Опыт программного макетирования**

Практическое исследование возможностей использования системы Stream Handler для разработки событийно управляемой архитектуры быстрого коллективного доступа к информационным ресурсам ДЗЗ было проведено путем создания прототипа активной

событийно управляемой системы, демонстрирующего принципиальные подходы к реализации описанного выше функционала. Блок-схема созданного прототипа в среде разработки Stream Handler Studio показана на *рис. 1*. Цифрами обозначены: 1 – блок формирования запроса на сервер Геофизической службы РАН, ГС РАН (<http://www.ceme.gsras.ru/>); 2 – блок обмена данными с сервером ГС РАН по протоколу HTTP; 3 – блок декодирования ответа сервера; 4 – блок отображения сообщений о событиях (землетрясениях); 5 – блок фильтрации значимых событий; 6 – прототип блока формирования реакции системы (команд обновления рангов значимости и генерации просмотровых продуктов); 7 – мониторы просмотра просмотровых продуктов (на текущей стадии разработки условно обозначены блоком отображения видеоданных).



*Рис. 1. Блок-схема прототипа разрабатываемой системы в среде Stream Handler Studio*

Общая схема функционирования приведенного прототипа такова. После запуска блок-схемы на исполнение в блоке 1 формируется запрос на сервер ГС РАН о времени и координатах наиболее поздних зарегистрированных землетрясений. Шаблон запроса с установленной периодичностью передается в блок 2 обмена данными с сервером по протоколу HTTP. Блок 2 передает запрос на сервер и получает автоматически сформированное сервером описание web-страницы, табличный список землетрясений, в формате HTML. Ответ сервера в виде единого текстового сообщения передается в блок декодирования 3, который выделяет в отдельные фрагменты, соответствующие записям по каждому отдельному событию. Набор записей собирается в единый *контейнер* и передается по двум путям: в блок визуализации форматированного текста 4 для отладки и контроля функционирования схемы (блок осуществляет вывод данных в показанное справа от него окно), а также в блок фильтрации значимых событий 5. Блок 5 осуществляет выборку значимых событий

по заданному в нем критерию (превышению порогового значения силы землетрясения) и передает координаты и время выделенных событий в блок 6 формирования реакции системы. Блок 6 регистрирует полученные данные для использования при формировании команд на обновление рангов значимости различных информационных ресурсов базы данных ДЗЗ, а также упреждающей генерации просмотревых продуктов и их отображения в соответствующих блоках (7 и т.д.). В результате развития настоящего прототипа блок 6 может быть преобразован в *комплекс* (вложенную блок-схему), и его функциональное наполнение будет осуществляться путем усложнения и настройки этого комплекса.

Рассмотрим более подробно способ реализации каждого из включенных в настоящий прототип блоков в системе Stream Handler. Блок 1 является стандартным блоком генерации текстовых пакетов TextCaptureCtrl. Блок настроен на периодическое формирование текстовых строк вида «[http://www.ceme.gsras.ru/cgi-bin/ccd\\_quake.pl?num=10&x=23&y=9](http://www.ceme.gsras.ru/cgi-bin/ccd_quake.pl?num=10&x=23&y=9)», интерпретируемых сервером ГС РАН как запрос на выдачу списка сообщений о землетрясениях в определенном табличном формате. Модификация запроса осуществляется редактированием указанной строки в настройках блока 1 в среде Stream Handler Studio или в сформированном файле сценария (скрипте) блок-схемы. При необходимости генерации нескольких запросов блок-схема может быть дополнена новыми индивидуально настроенными блоками типа блока 1.

Блок 2 обмена данными с сервером по протоколу HTTP не имеет стандартной реализации в библиотеках системы Stream Handler и был реализован путем дополнительного программного моделирования. Был описан новый тип блока HttpRequest, его экземпляр был встроен в исполняемую блок-схему рассматриваемого прототипа. Блок имеет следующие настройки: URL сервера, параметры запроса и признак наличия входного контакта (при наличии входного контакта URL сервера и параметры запроса извлекаются непосредственно из входных данных, начальные установки игнорируются).

Блок 3 реализован на базе стандартного блока MatchTemplate системы Stream Handler. Он выделяет из входного текста фрагменты, соответствующие заданному шаблону (с учетом метасимволов). В данном случае был применен шаблон, позволяющий разобрать представление одной табличной записи списка зарегистрированных землетрясений, формируемого сервером ГС РАН в формате HTML и заполнить выходную структуру данных блока соответствующими параметрами.

Блок 4 является стандартным блоком TextRender визуализации форматированного текста системы Stream Handler. Визуализация данных в рассмотренном примере осуществляется в отдельном окне Float window 1, показанном на *рис. 1*.

Блок 5 фильтрации значимых событий реализован в настоящем прототипе средствами программного моделирования. Он разбирает содержимое каждой строки поступающего списка и выделяет данные о дате (первый параметр), времени (второй параметр), широте и долготе (третий и четвертый параметр), интенсивности (предпоследний параметр), а также глубине очага землетрясения и других параметрах. Набор параметров формирует единый *контейнер*, передаваемый на обработку в следующие блоки при условии, что интенсивность землетрясения превысила некоторый заданный порог (7 по умолчанию).

Блок 5 является результатом первого пробного подхода к реализации фильтров значимых событий. При дальнейшем развитии системы предполагается изучить и реализовать более универсальные и гибкие способы фильтрации значимых событий с использованием стандартных средств Stream Handler и дополнительного программного моделирования.

О блоке 6 было сказано при описании общей схемы функционирования прототипа.

В качестве средств локальной визуализации подготовленных просмотрных продуктов (в целях тестирования и отладки) предполагается использовать широкий набор стандартных блоков отображения системы Stream Handler. Включение блока отображения в реализуемую систему условно обозначено блоком (7) на блок-схеме разработанного прототипа (*рис. 1*). Следует отметить, что разработанный прототип протестирован и отлажен в реальном сетевом окружении. *Рис. 1* отображает стадию его работы после приема очередной порции данных от сервера ГС РАН, декодирования и отображения списка соответствующих сообщений о последних произошедших землетрясениях.

### Заключение

В настоящей работе проанализирована принципиальная возможность реализации событийно управляемой архитектуры быстрого коллективного доступа к информационным ресурсам ДЗЗ на базе системы Stream Handler, а также описан созданный программный прототип, работающий в условиях реального сетевого окружения. Результаты работы позволяют сделать следующие основные выводы.

1. Базовые принципы, заложенные в систему Stream Handler, позволяют реализовать разрабатываемую событийно управляемую, многопоточную архитектуру эффективными, надежными, быстрыми и наглядными методами программирования; совокупность основных понятий системы Stream Handler, типов данных и стандартного функционала соответствует предметной области решаемой задачи.

2. Разработка общего решения, а также программное моделирование отдельных специфических узлов обработки существенно упрощены в среде разработки Stream Handler Studio (по сравнению с полным циклом разработки таких решений с этапа проектирования) как за счет средств визуального программирования, так и в связи с особенностями самой системы, ориентированной на обработку потоковых данных.

3. Модернизация и развитие проекта может осуществляться непосредственно в ходе ее оперативного функционирования; при этом имеются средства для надежного и эффективного перехода от старых версий алгоритма к новым.

4. Разработанный и протестированный в реальном сетевом окружении прототип активной событийно управляемой системы подтвердил предварительные результаты проведенного анализа возможности решения поставленной задачи на базе системы Stream Handler; указаны пути и средства развития прототипа до полнофункционального модуля.



## Благодарности

Авторы благодарят компанию «The File X Ltd», Ness Ziona, Israel за предоставленное программное обеспечение. Работа выполнена при поддержке грантов РФФИ № 11-07-12028-офи-м-2011, 13-07-00513.

## Литература

1. *Savorskiy V., Tishchenko Yu.* Event-driven information system designed for emergency applications // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B8. Beijing, July 2008. P. 241–248.
2. *Savorskiy V., Tishchenko Yu.* Upgrade of space data information system to use in natural disaster emergency // ISRSE-33, Stresa, Italy, May 4–9, 2009, ref.350 (flash card publication).

### **The implementation of event-driven architecture for rapid collective access to the information resources of the Earth remote sensing based on the Stream Handler technology**

**D.M. Ermakov<sup>1,2</sup>, K.S. Emelyanov<sup>3</sup>, V.P. Savorskiy<sup>1</sup>, A.P. Chernushich<sup>1</sup>**

<sup>1</sup> *Institute of Radioengineering and Electronics of RAS, Fryazino Department, Fryazino, Russia*  
*E-mail: dima@ire.rssi.ru*

<sup>2</sup> *Space Research Institute of RAS, Moscow, Russia*

<sup>3</sup> *Research Center for Earth Operating Monitoring, Moscow, Russia*

Creating information systems to provide fast delivery of relevant information on areas of natural disasters in a convenient form for potential users is one of the most important tasks of satellite monitoring of the Earth. Presented are the results of software prototyping of this kind of system based on the Stream Handler technology. The important advantages of Stream Handler are the high level of the implemented solutions in the form of block diagrams and the ability to modify (on-demand) and self-modify (as a response to certain “trigger” events) the processing algorithm at run time. The reports of seismic events were used as the triggering ones. The software prototype was realized as an integrated service of the CPSSI of FIRE RAS, targeting on usage of information resources of typical Russian centers of space data.

**Keywords:** distributed information systems, event-driven architecture, Stream Handler.