

# EKATRA – промышленная SCADA с потоковой передачей данных на корпоративный уровень

Первоначально созданная для контроля объектов ядерной энергетики, промышленная SCADA-система EKATRA выросла до масштабов корпоративной платформы, которая может быть единым источником данных по всем аспектам промышленной автоматизации для пользователей и прикладных задач компании любого масштаба.

**С. Волошенюк, О. Зиновец, И. Клименко, Л. Тараненко, ivl@ivl.ua**

**С**истема EKATRA обеспечивает в реальном времени сбор данных от систем промышленной автоматизации, промышленных контроллеров, датчиков и исполнительных механизмов, доставку данных по сетям и каналам связи в системы управления производством, системы управления предприятием, корпоративные информационные системы, приложения анализа больших объемов информации (Big Data) и другие.

EKATRA собирает, обрабатывает и накапливает данные от оборудования и систем вне зависимости от места их размещения, обеспечивает управление, защиту, контроль, а также потоковую передачу данных в системы корпоративного уровня.

## Предпосылки создания

EKATRA является промышленной SCADA-системой реального времени, которая создавалась для поддержки персонала четырех атомных станций Украины (всего 15 энергоблоков с установленной мощностью 13 835 МВт), а также Дирекции государственной компании «Энергоатом» – оператора этих АЭС. Конечными пользователями системы являются также Госатомрегулирование Украины и Государственный научно-технический центр ядерной и радиационной безопасности, обеспечивающий научное сопровождение регулирующего органа. Кроме того, система обеспечивает данными Украинскую базу данных надежности и систему RODOS, которая предназначена для оценки в режиме реального времени последствий ядерных или радиационных аварий на АЭС.

Система осуществляет непрерывно (24/7) сбор, обработку и накопление данных от систем контроля и управления АЭС, общестанционных систем, серверов, ПЛК, отдельных датчиков, размещенных в помещениях на площадках АЭС и за их пределами (всего около 300 тысяч значений параметров, частота обновления – 1 раз в секунду, при необходимости частота обновления может быть увеличена).

Подсистема хранения данных обеспечивает ведение архивов значений принимаемых параметров, баз данных метаинформации о параметрах, справочной информации, ведения журнала истории событий, информации о системе и пользователях системы.

Представление информации пользователям осуществляется с помощью человека-машинного интерфейса (ЧМИ), который построен как система видеокадров со встроенными средствами навигации. ЧМИ позволяет пользователю контролировать ход технологического процесса и состояние безопасности, включая:

- ▶ общий обзор системы и контролируемых параметров;
- ▶ обработку информации в реальном времени;
- ▶ тренды технологического процесса;
- ▶ аварийные сигналы и обнаружение неисправностей;
- ▶ состояние серверов, ПЛК, исполнительных механизмов, вычислительной сети;
- ▶ техническое обслуживание и устранение неисправностей.

Пользователю предоставлена возможность выбора способа представления информации (динамические мнемосхемы, таблицы, графики),

метода обработки данных, получения справочной информации (нормативные, технические документы и пр.).

Серверы системы, установленные на АЭС, обмениваются данными с серверами корпоративного уровня ГП «НАЭК «Энергоатом» по существующим арендованным каналам наземной и космической связи. Пользователям системы на любой из площадок доступны любые данные, которые хранятся в системе, вне зависимости от того, где физически эти данные расположены.

Система является независимой надстройкой над системами управления и контроля АЭС, что дает возможность ее беспрепятственной модификации, масштабирования без вмешательства в работу АЭС.

Накопление в системе огромных объемов данных, которые порождаются на АЭС, предоставляет техническим экспертам и руководству компании новые дополнительные возможности по их обработке и получению такой оперативной и бизнес информации, которая ранее была недоступна. Таким образом, EKATRA доведена до уровня полноценной SCADA, которая, в то же время, обладает рядом дополнительных конкурентных преимуществ.

## Исходные условия

АЭС, как источники данных для EKATRA, имеют ряд особенностей, которые характерны также и для других отраслей промышленности. Эти особенности, а также требования, предъявляемые заказчиком, определили выбор технологий и решений при создании EKATRA.

▶ **Огромные потоки данных.** В систему поступает более 300 000 значений контролируемых параметров в секунду.

▶ **Разнотипные источники данных.** Источниками данных являются большое количество разнотипного оборудования, систем, приложений от различных производителей, которые используют разнообразные протоколы и интерфейсы. Система должна принять данные от всех источников, преобразовать их в свои форматы и иметь возможность обрабатывать их по собственным правилам.

▶ **Необходимость воспроизведения исходных видеокадров ЧМИ.** Для решения задач контроля технологического процесса, поддержки операторов в аварийных ситуациях на рабочих местах АЭС и Дирекции ГП «НАЭК «Энергоатом» необходимо обеспечить воспроизведение видеокадров блочных щитов управления, систем радиационного контроля, общестанционных систем. В одной системе должны быть собраны и воспроизведены видеокадры разных систем, имеющих собственные отличные друг от друга языки описания видеокадров, базы дан-

ных, состав и форматы метаинформации.

#### ▶ **Большое количество видеокадров ЧМИ.**

ЧМИ. Количество видеокадров для одного энергоблока может достигать 500. Их ручное воспроизведение и поддержание в актуальном состоянии в системе займет неприемлемо длительное время. Конвертация видеокадров первичных систем в SCADA должна быть автоматизирована.

#### ▶ **Необходимость создания дополнительных видеокадров ЧМИ.**

Система должна обеспечивать обработку исходных данных, решение на ее основе задач поддержки нормальной эксплуатации и аварийного реагирования, которые не были предусмотрены проектом АЭС, и отображать результаты на рабочих местах персонала. Таким образом, система должна иметь собственную среду исполнения, отличную от среды исполнения первичной системы, в которой могут присутствовать как исходные видеокадры, так и вновь созданные.

#### ▶ **Большое количество площадок.**

Система должна обеспечить сбор данных с 4 площадок АЭС и предоставить информацию пользова-

телям на 10 площадках, распределенных по территории площадью более 600 000 км<sup>2</sup>.

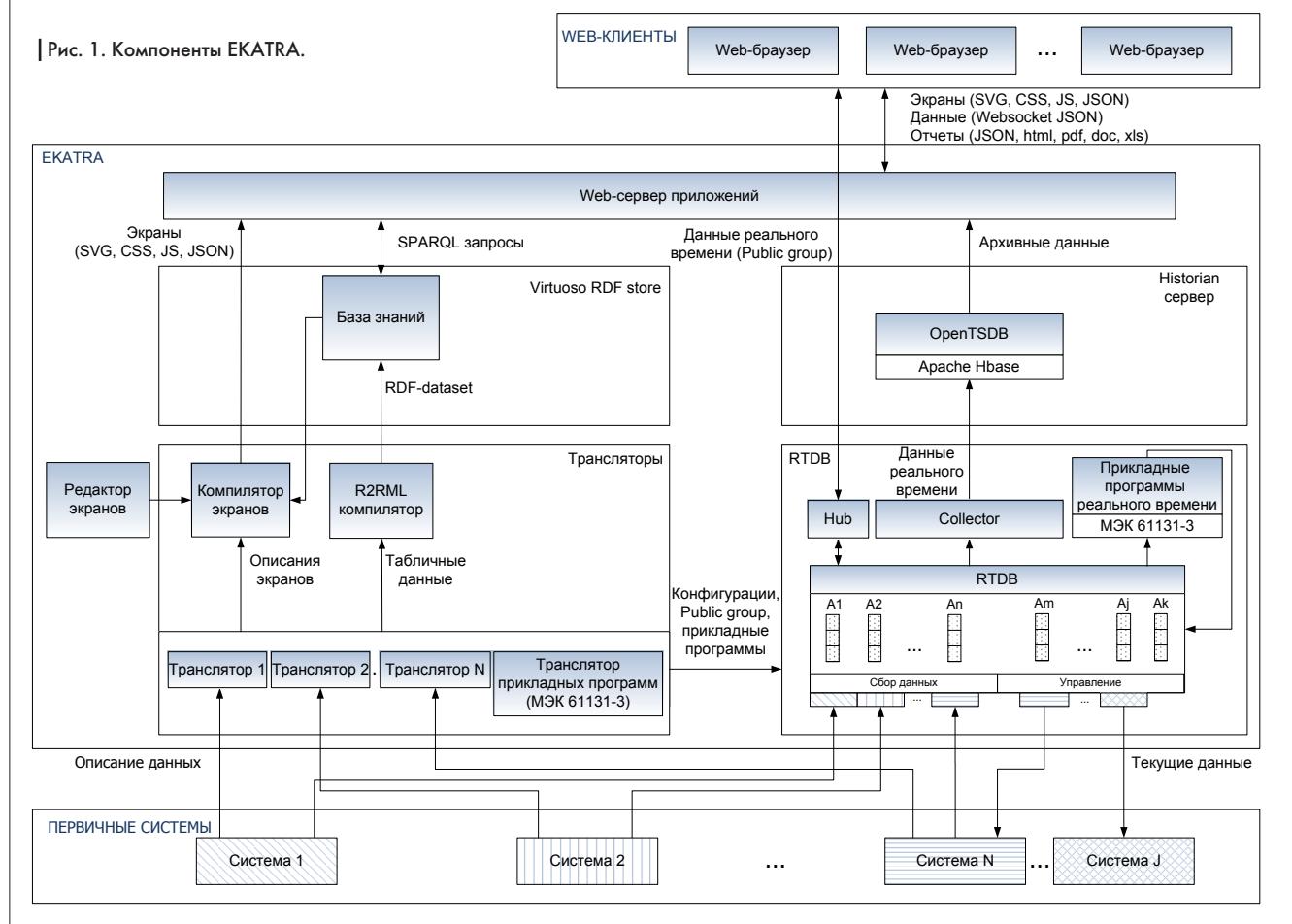
#### ▶ **Большое количество пользователей.**

На площадке каждой из АЭС количество пользователей достигает 100 человек. При этом система, учитывая рост ее востребованности, должна быть готова к значительному росту их числа.

#### ▶ **Сложность сопровождения системы.**

Должна быть обеспечена синхронизация информации в базах данных первичных систем и базах данных EKATRA на всех площадках. Данные АЭС подвержены относительно частым изменениям, которые вызваны модернизацией систем контроля и управления, изменениями, которые вносятся в технологические системы в ходе технического обслуживания и ремонтов. Соответствующие изменения должны отслеживаться и немедленно вноситься в структуру данных EKATRA. Учитывая количество площадок и источников данных, решить эту задачу чисто организационными методами (общением персонала АЭС и системных администраторов на площадках) невозможно. Требуются средства,

| Рис. 1. Компоненты EKATRA.



которые обеспечат автоматизацию внесения изменений.

- ▶ **Требования к надежности.** Согласно требованиям нормативных документов Украины, коэффициент готовности системы должен быть не менее 0,99, время восстановления – не более 60 минут.

### Используемые технологии

Программное обеспечение EKATRA построено на основе WEB-технологий. Большая часть применяемых компонентов является высококачественными продуктами с открытым исходным кодом, компоненты которых прошли многократную проверку в Интернет-сообществе.

#### ▶ **OpenLink Virtuoso Universal Server**

используется в системе в качестве RDF хранилища/базы знаний о объекте мониторинга.

#### ▶ **OpenTSDB** – масштабируемая база данных временных рядов, используется в качестве сервера истории.

#### ▶ **Apache Hbase** – распределенная NoSQL база данных, используется в качестве хранилища временных рядов OpenTSDB.

#### ▶ **RabbitMQ** – сервер обмена сообщениями, используется в качестве связующего ПО/среды обмена сообщениями в системе.

#### ▶ **StackStorm** – управляемое событиями средство автоматизации, используется для автоматического устранения неполадок, обеспечения безопасности, облегчения поиска неполадок, инсталляции программного обеспечения и др.

#### ▶ **MapServer** – серверная геоинформационная система.

#### ▶ **PostgreSQL** – реляционная база данных.

При разработке системы использовались объектно-ориентированные языки программирования C++, Python, JavaScript.

В отличие от большинства SCADA-систем, в которых вся функциональность реализована в рамках одного большого процесса, серверное ПО EKATRA разработано, как распределенная система микросервисов, взаимодействующих по строго документированным API протоколам. Микросервисы обеспечивают масштабируемость и отказоустойчивость системы, поскольку каждый микросервис может быть многократно воспроизведен и нагрузка по обработке информации может быть распределена между множеством таких микросервисов. Отказ отдельного микросервиса не вызовет аварийной остановки системы в целом.

Технология микросервисов позволяет строить произвольно масштабируемые системы, без ограничения, как по количеству входных данных, так и по количеству клиентов. Разработчикам доступно огромное количество отлаженного, готового к использованию отказоустойчивого ПО в исходных кодах.

Поскольку микросервисы являются маленькими и простыми, они подлежат простой отладке, могут разрабатываться и модифицироваться независимо друг от друга. Любой микросервис можно остановить, обновить и снова запустить без остановки всей системы.

Все микросервисы EKATRA упакованы в контейнеры Docker, что гарантирует работу приложения на любом другом компьютере с операционной системой Linux независимо от каких-либо настраиваемых параметров. Кроме того, контейнеры Docker изолируют приложения друг от друга и от базовой инфраструктуры системы. По умолчанию Docker обеспечивает мощную изоляцию приложения, ограничивая влияние любой возникшей проблемы в пределах одного контейнера, не позволяя оказать влияние на весь компьютер.

Использование технологии контейнеров позволяет создавать самодостаточные легко портируемые приложения, автоматизировать процессы сопровождения ПО. Это минимизирует операции по настройке системы, требования к квалификации сопровождающего систему персонала, а также позволит при необходимости в будущем с минимальными усилиями перенести работу серверного ПО во внешние data-центры.

### База данных реального времени (RTDB)

Предназначена для централизованного хранения текущих значений измеряемых или расчетных параметров и организации доступа к ним для последующей обработки и визуализации. Характерной особенностью значительного числа этих параметров является быстрое изменение их значений.

RTDB представляет собой набор именованных массивов (структур) данных в разделяемой памяти, доступной различным процессам сервера. Реализует модель доступа к данным «издатель-подписчик».

Поставщики информации (издатели):

- ▶ получают текущие значения динамических параметров (от датчиков, внешних систем и т.д.);

- ▶ обновляют массивы RTDB;
- ▶ уведомляют потребителей об этих изменениях.

Потребители информации (подписчики):

- ▶ обновляют значения на мнемосхемах пользовательских экранов;
- ▶ сохраняют изменения на серверах истории;
- ▶ выполняют роль сенсоров для сервера автоматизации, производя StackStorm-триггеры;
- ▶ реплицируют изменения в массивах на удаленные сервера.

### База данных метаинформации

Предназначена для хранения редко меняющихся данных. Содержит:

- ▶ информацию о контролируемых параметрах (наименование, единицы измерения, уставки, адрес в RTDB (имя массива и индекс в массиве) и т.д.);
- ▶ произвольную дополнительную информацию (может содержать, например, географические координаты точек измерения);
- ▶ информацию об объектах прикладной области.

Реализована в виде базы знаний RDF-хранилища сервера приложений Virtuoso. Поддерживает язык запросов SPARQL. Не требует проектирования структуры базы данных.

### Сервер истории

Сервер истории предназначен для долговременного хранения параметров, зависящих от времени. Содержит:

- ▶ коллектор – подписчик RTDB информации, записывающий данные пары «время-значение» в хранилище временных рядов;
- ▶ базы данных временных рядов – специализированное хранилище совокупности пар «время-значение», которое реализовано с помощью OpenTSDB.

В качестве физического хранилища записей истории используется Apache Hbase, что обеспечивает отказоустойчивый способ хранения больших объемов данных.

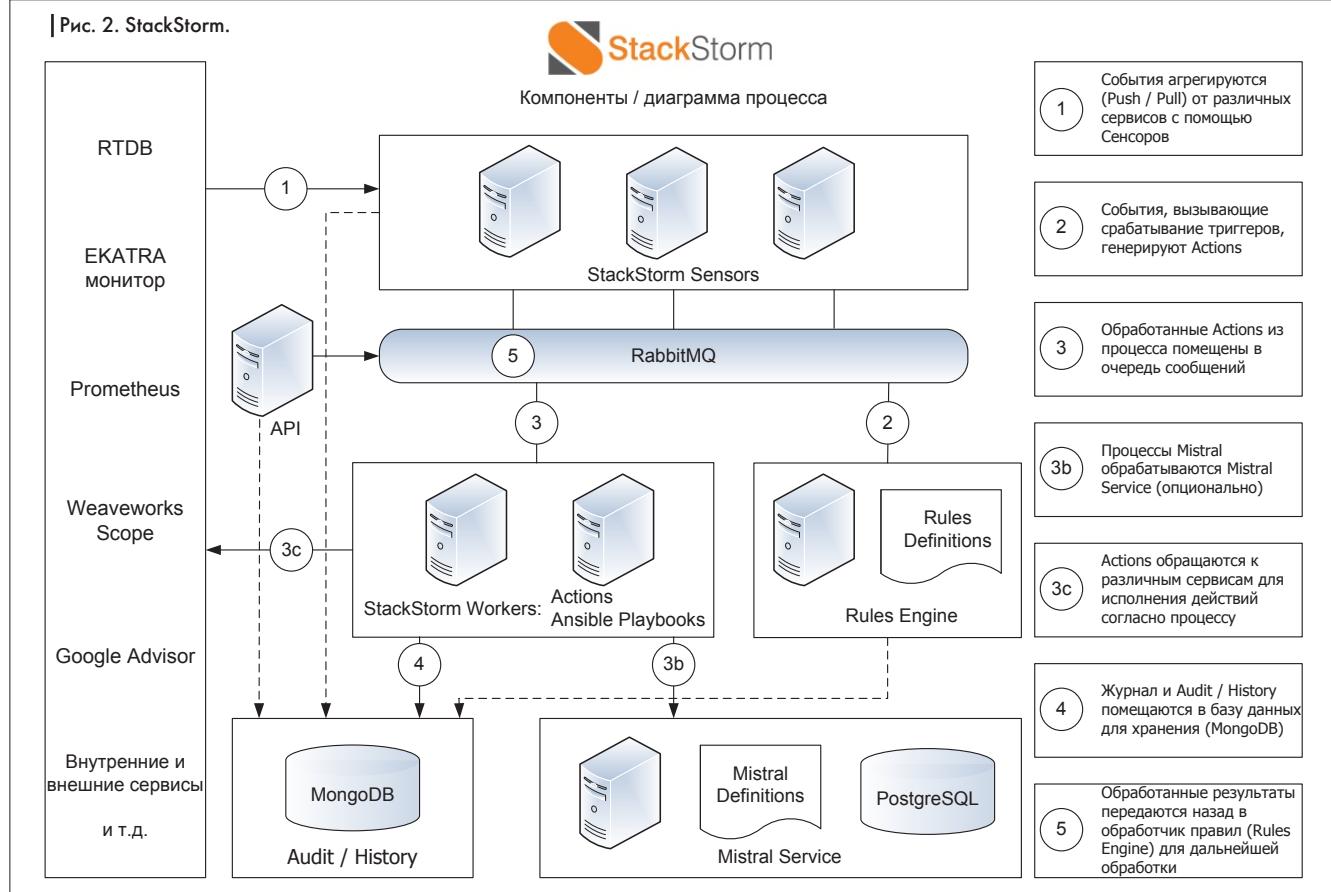
### Web-сервер приложений

Web-сервер приложений предназначен как для организации человеко-машинного интерфейса с системой, так и для обеспечения машино-машинного взаимодействия (RPC).

### Сервер автоматизации

Сервер автоматизации предназначен для обеспечения реактивного характера поведения системы. Его

Рис. 2. StackStorm.



задача – автоматически детектировать исходные события и автоматически выполнять предусмотренные для этих событий процедуры. Сервер реализован на платформе StackStorm, его функционирование основано на правилах.

- Сервер автоматизации включает:
- ▶ Триггеры (Triggers), представляющие внешние события.
  - ▶ Действия (Actions) – процедуры, которые могут быть выполнены сервером автоматизации. Действия могут быть вызваны непосредственно пользователем или автоматически, как реакция на внешние события.
  - ▶ Правила (Rules) – определяют критерии выполнения действий при возникновении триггеров/внешних событий.
  - ▶ Рабочие процедуры (Workflows) – сложные действия, состоящие из элементарных действий и определяющие порядок, условия перехода и передачу данных между шагами/атомарными действиями. Рабочие процедуры содержатся в библиотеке действий, и могут быть вызваны вручную или запущены в соответствии с правилами.
  - ▶ Аудит – логирование выполнения действий, запускаемых вручную или автоматически, содержит пол-

ную информацию о контексте запуска и результатах выполнения.

Сервер автоматизации обеспечивает:

- ▶ аварийные сообщения;
- ▶ отправку сообщений на включенные компьютеры пользователей, в том числе при помощи электронной почты и SMS;
- ▶ организацию сложных рабочих процедур (Workflow), включающих части, автоматически выполняемые ПО, а также части, которые выполняются с участием пользователей системы.

#### WEB-приложение Brama

Web-приложение Brama обеспечивает:

- ▶ управления конфигурацией ПО,
- ▶ управления правами пользователей системы,
- ▶ управление обновлением программного обеспечения системы,
- ▶ мониторинг состояния оборудования и программного обеспечения системы.

#### Сбор данных

Сбор данных обеспечивается с помощью базы данных реального времени (RTDB). Назначение RTDB – получение данных в реальном времени. ПО RTDB обеспечивает взаимодействие с первичными системами.

Поскольку каждая первичная система может иметь собственный формат и протокол передачи данных, RTDB включает различные программные шлюзы. Назначение этих шлюзов – преобразовать данные, получаемые от первичных систем, в унифицированный формат. Далее в системе используются унифицированные массивы данных.

Первичные системы, существующие стандарты (OPC, DDS, OPC-UA), SCADA, как правило, описывают объект (источник данных) в целом. Т.е. в общей базе данных одновременно находятся статические данные (паспорт параметра): наименование, идентификатор, единица измерения, уставки, – и собственно значения параметра.

В системе EKATRA текущие значения и статическая информация (метаданные) разделены и хранятся в разных базах данных. В RTDB хранятся только текущие значения. Сбор данных от систем-источников данных (первичных систем), обеспечивается в режиме «on-line».

Процесс сбора данных обеспечивает специальные программные модули. Принятые данные размещаются в разделяемой памяти (Shared memory) программного модуля. Использование этого механизма обеспечивает высокую

скорость доступа к текущим данным со стороны других процессов системы.

Данные в памяти организованы в виде массивов, в которых содержится информация о типах хранимых данных, размерах, времени последнего поступления, принятые значения аналоговых и/или дискретных параметров первичных систем. В RTDB данные описываются следующим образом: имя массива и индекс в массиве.

Потребителями информации, полученной от первичных систем, являются процессы, которые выполняют отображение и динамическое обновление данных на страницах (видеокадрах), сохранение информации в архивах системы, передачу данных во внешние системы, мониторинг процессов сбора данных.

### Обработка метаданных

Для хранения метаинформации в системе используется RDF (Resource Description Framework) модель представления данных, разработанная Консорциумом W3C. RDF-хранилище позволяет одновременно хранить как данные произвольной структуры, так и описатели этих данных, т.е. собственно структуру самой базы данных.

В RDF-хранилище каждый объект доступен и связан с другими объектами через URL (согласно технологии Semantic Web). Для каждого объекта в хранилище определены место и способ размещения, правила доступа к объекту.

RDF-хранилище EKATRA обеспечивает хранение и унифицированный доступ как к информации, полученной от первичных систем, так и к информации, характеризующей саму систему. На вход от внешних систем поступают табличные (реляционные) данные. В рамках EKATRA создан компилятор, который реализует стандарт R2RML (Relational Data to Model Language), на котором описываются преобразования табличных данных в RDF-факты.

Компилятор SQL позволяет для любой комбинации таблиц: Dbf, Oracle, Microsoft Excel, JSON-массивы и т.д. сделать операцию Joint, и обработать сводную таблицу компилятором R2RML, получив в результате данные в формате RDF. Преобразованные и обработанные данные поступают в RDF-хранилище.

Данные, находящиеся в RDF-хранилище, могут быть запрошены и обновлены с помощью языка запросов SPARQL. Для доступа к информации RDF-хранилища используются Web-сервисы.

RDF-хранилище может иметь распределенную структуру и размещаться на серверах, расположенных на территориально удаленных площадках. RDF-хранилище EKATRA реализовано с помощью ПО OpenLink Virtuoso Universal Server, которое поддерживает масштабирование и кластеры серверов.

### ЧМИ

Для представления информации пользователям системы используются стандартные интернет-браузеры. Для создания видеокадров используются современные технологии представления данных (Scalable Vector Graphics (SVG), Cascading Style Sheets (CSS), Extensible Markup Language (XML)).

На вход системы поступают видеокадры, язык описания которых является индивидуальным для каждой из первичных систем. Для каждого из языков описания разрабатывается свой транслятор, который преобразует описания видеокадров в язык описания EKATRA. После первичного преобразования в системе присутствуют видеокадры от всех первичных систем в едином представлении, что упрощает процесс сопровождения системы.

Скомпилированные видеокадры в EKATRA представлены в формате SVG, CSS. SVG – это низкоуровневый язык масштабируемой векторной графики, который требует указания координат всех отображаемых точек. Язык EKATRA позволяет рассчитывать координаты точек видеокадра. Координаты точек описываются в виде формул и геометрических преобразований. Язык EKATRA содержит в себе такое понятие, как «макро», что позволяет разработать сложный объект и повторно, многократно его использовать, корректируя при использовании, если необходимо.

Описание в формате Cascading Steel Sheet содержит информацию, определяющую внешний вид видеокадра в браузере. После того, как описания видеокадров преобразованы в язык описания EKATRA, файл на этом языке компилируется в формат SVG. Необходимая для компиляции метаинформация поступает из RDF. Этот процесс унифицирован для всех первичных систем.

Компилятор для каждого видеокадра создает таблицу с метаинформацией группы параметров, которая содержит следующие поля: наименование параметра, единица измерения, уставки. Такие таблицы используются для оптимизации запросов

к хранилищам данных при каждой визуализации видеокадра.

Для каждого видеокадра при компиляции создается JSON «public» группа с метаинформацией об отображаемых параметрах и «public» группа для RTDB, в которой для каждого параметра указывается индекс массива и индекс в массиве.

При помощи языка описания видеокадров возможна разработка собственных дополнительных видеокадров. Поскольку в рамках EKATRA функционирует единая RTDB, то на создаваемых видеокадрах могут одновременно отображаться данные от нескольких первичных систем или даже территориально удаленных объектов, что было невозможным для каждой из первичных систем.

HMI представляет:

- ▶ мнемосхемы первичных систем и вновь созданные экраны, с отображением статических и динамически изменяющихся элементов,
- ▶ текущие, архивные данные в виде таблиц и графиков,
- ▶ расчетные величины,
- ▶ систему навигации по страницам.

Для создания видеокадров предоставляются готовые библиотеки шаблонов графических элементов. Библиотеки при необходимости могут расширяться разработчиками.

### Среда исполнения

Основными компонентами среды исполнения EKATRA являются WEB-сервер приложений, база данных реального времени, база знаний и база временных рядов, которые функционируют на сервере, а также браузер на рабочем месте пользователя. WEB-сервер приложений получает статические данные из базы знаний и отдает их браузеру, генерирует отчеты по шаблонам и через браузер предоставляет их пользователю.

Браузер получает от WEB-сервера статическую информацию, которая включает библиотеки Java Script, видеокадры в SVG формате, описания параметров, которые представлены на видеокадре, в JSON формате. Для получения динамических данных создается канал связи браузера с RTDB по WebSocket (WS). Получение данных осуществляется при помощи технологии «издатель-подписчик». Подписка оформляется на «public» группу значений, отображаемых на видеокадре. После получения данных браузер выполняет обновление видеокадра.

Для получения архивных данных браузер обращается к WEB-серверу приложений, указав ID параметров

# ЕКАТРА

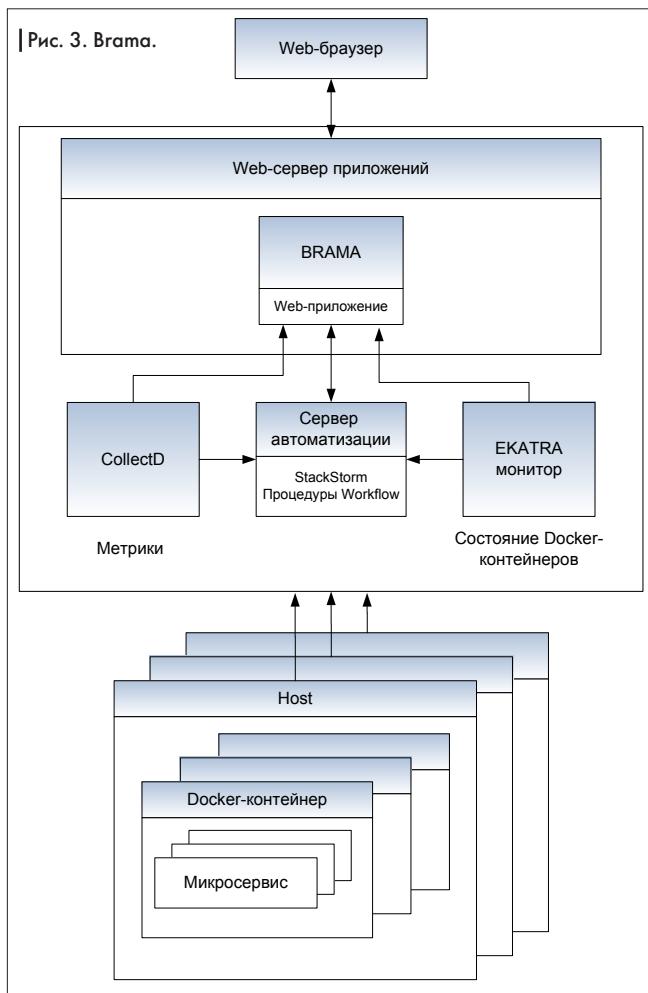
Ти будеш знати про свої об'єкти стільки,  
скільки не зневідомив ніколи



«ЕКАТРА» в перекладі з санскриту – «разом»



ТОВ «ІВЛ Обладнання та інжиніринг»  
Тел.: (044) 499 2132  
Факс: (044) 499 2134  
e-mail: ivl@ivl.ua  
<http://ivl.ua>



и заданный интервал выборки. WEB-сервер, в свою очередь, обращается к OpenTSDB, который формирует временные ряды по заданному условию выборки.

#### Прикладные задачи реального времени

EKATRA позволяет вести в реальном времени обработку поступающих данных. Прикладная программа обработки данных может считывать необходимые данные из RTDB, выполнять расчеты и записывать в RTDB результаты своих расчетов. Далее, эти результаты могут обрабатываться наравне с данными, поступающими от первичных систем.

Прикладные алгоритмы реализуются в специальной среде разработки, которая позволяет использовать языки стандарта МЭК 61131-3 (FBD, SFC, IL, LD, ST). Среда разработки включает транслятор с языков МЭК 61131-3 в код языка, который используется в ПО EKATRA. Среда разработки обеспечивает доступ к информации в RDF-хранилище. Соответствие стандарту МЭК 61131-3 снижает требования к разработчику, не требуя специальных знаний в области программирования.

#### Управление

Команды управления от Web-клиентов или прикладных программ реального времени выдаются на системы нижнего уровня через RTDB.

#### Архивы

RTDB хранит только текущие и предыдущие значения поступающих данных. Хранение данных обеспечивает OpenTSDB (Time Serial Data Base). В системе реализован коллекtor данных, который подписывается на получение данных RTDB. Задача коллектора сократить объем хранимой информации

путем аппроксимации временного ряда кусочно-непрерывной функцией с переменным шагом между точками.

На выходе коллектор выдает событие: идентификатор (ID) параметра, время, значение параметра. Событие попадает в OpenTSDB. По сути, OpenTSDB представляет собой API к временным рядам. TSDB может получать события по точкам временного ряда, сохранять их и выполнять запросы на выдачу временного ряда параметра с заданным ID за заданный интервал времени.

С целью оптимизации процесса отображения при запросе временного ряда за длительный период обеспечивает агрегирование данных. OpenTSDB выдает среднее, максимальное, минимальное значения за укрупненные периоды, из которых можно строить график за месяц, за год или более длительный период. Количество выдаваемых точек может быть согласовано с разрешением экрана монитора.

Для хранения данных в OpenTSDB используется Apache HBase. HBase позволяет организовывать облако серверов хранения, которое можно расширять динамически без необходимости переконфигурирования системы.

#### Администрирование и сопровождение EKATRA

Для контроля состояния системы используется программный продукт CollectD, который с заданным интервалом собирает статистику об использовании ресурсов системы. Для контроля за микросервисами, Docker контейнерами, их состоянием, потреблением памяти и процессорного времени используется EKATRA-монитор.

Основным продуктом, на котором базируется система администрирования и сопровождения EKATRA, является StackStorm. На этом продукте построена, как реакция системы на события по мониторингу, так и все процедуры инсталляции и обновления ПО. StackStorm используется также и в процессе разработки системы. Процесс генерации нового кода, система контроля версий, запуск компиляторов, сборка, создание тестовой среды, отладка – все это автоматизировано с помощью StackStorm.

Данные CollectD, EKATRA монитора, RTDB поступают на вход сенсоров StackStorm. На основании правил, которые хранятся в базе знаний StackStorm, могут генерироваться триггеры, т.е. события на которые StackStorm может реагировать. Правила определяют, какие действия (actions) будут вызваны как реакция на триггеры/события.

В зависимости от типа события и его содержания, реакция StackStorm может быть направлена на управление потоком данных от RTDB, управление вычислительными ресурсами системы, запуск/останов того или иного микросервиса, запуск/останов компилятора и др.

Действия (actions), выполняемые StackStorm, могут быть простые (выполнение отдельных команд Shell) или целевые последовательности действий (Workflow).

Workflows могут включать последовательность простых действий (actions), выполняемых автоматически, или включать также действия персонала, которые могут выполняться вручную и со значительной временной задержкой. Для отложенных во времени действий используется программный продукт Mistral. Все действия фиксируются в архиве StackStorm и могут быть просмотрены с помощью сервиса Audit.

Для обеспечения человека-машинного взаимодействия с EKATRA используется собственное WEB-приложение Brama. Brama обеспечивает удобный интерфейс, приспособленный к решаемым EKATRA прикладным задачам. Через WEB-приложение Brama обеспечивается взаимодействие администратора EKATRA с StackStorm, CollectD, EKATRA-монитором (рис. 3). **MA**