

ее применение позволит получать необходимые данные напрямую от лиц участвующих в работе предприятия,

минимизировав тем самым участие группы аналитиков на данном этапе проектирования ИС.

#### Список литературы:

1. **Рогозов Ю.И., Свиридов А.С., Красноруженко Д.А.** Методология разработки баз данных предметной области на основе знаний конечного пользователя // Технологии разработки информационных систем, том 1, Таганрог 2011г., – с 131-136
2. **Карпова Т.С.** Базы данных. Модели, разработка, реализация; СПб: Питер, 2001. – 304 с.
3. **Свиридов А.С.,** Методика проведения предпроектного обследования с целью проектирования информационной сети предприятия, Телекоммуникации. 2004. № 4. С. 27.
4. **Рогозов Ю.И., Бутенков С.А., Свиридов А.С., Горбань Н.С., Дубровский А.А., Друппов С.А., Жибулис Ю.А., Почечуев Н.В., Стукотий Л.Н.,** Метод Создания Инструментальных Средств Разработки Автоматизированных Информационно-Управляющих Систем, Мехатроника, автоматизация, управление. 2008. № 1. С. 52.

УДК 004.434

# ЭНЕРГОВАТЧЕР: ДИНАМИЧЕСКИ НАСТРАИВАЕМАЯ СИСТЕМА МОНИТОРИНГА ЭНЕРГОПОТРЕБЛЕНИЯ

Е.А. Калашников, Л.Н. Лядова

## Введение

Эффективность использования топлива и электроэнергии в России пока остается крайне низкой, а вместе с тем, происходит подорожание энергоресурсов, что приводит к увеличению их доли в себестоимости продукции для многих промышленных предприятий. Таким образом, современные условия диктуют необходимость радикального изменения отношения к организации энергоучета, оптимизации энергопотребления. Эти задачи актуальны не только для промышленности, но и для других областей деятельности, жизнеобеспечения. Их решение требует использования специального оборудования, установки и настройки приборов, позволяющих, как минимум, контролировать

расход ресурсов. Получаемые от этих приборов данные – основа для выполнения анализа параметров, оптимизации энергопотребления. Специальное программное обеспечение позволяет автоматизировать сбор и обработку данных, решение оптимизационных задач.

Многие фирмы, работающие в этой области, представляют на рынке собственные программно-аппаратные комплексы, предназначенные для решения указанных задач. Существуют также разработки сторонних фирм, занимающихся производством программного обеспечения, позволяющего создавать решения, интегрирующие оборудование разного типа от различных фирм-производителей в единую систему. Эти работы ведутся как на уровне от-

дельных объектов, предприятий и организаций, так и на региональных уровнях.

Поставляемые разработчиками оборудования программы чаще всего не дают возможности интеграции данных, получаемых из множества источников разных типов. Создание комплексных систем мониторинга и учета энергопотребления – это актуальная задача, которая предусматривает разработку программно-аппаратных комплексов, предназначенных для решения взаимосвязанных задач сбора, обработки и архивации данных о функционировании сложных систем, включающих различные источники и потребителей энергоресурсов.

### 1. Системы мониторинга: архитектура и подходы к реализации

Несмотря на многообразие существующих решений можно выделить общие для всех функции, подходы к реализации.

Возможны два основных подхода к разработке систем мониторинга – «ручной» (система программируется с нуля под конкретную архитектуру аппаратного комплекса системы и, следовательно, обладает оптимальными для этих условий характеристиками производительности, но проигрывает в возможности адаптации) и автоматический (программный код и базы данных генерируются на основе построенной модели системы).

Второй подход является стандартным при использовании для разработки CASE-средств. При этом происходит трансформация исходной модели системы мониторинга в её программный код. В этом случае мы обладаем универсальным решением с высокой скоростью разработки в ущерб производительности генерируемого кода. Однако и это решение тоже не является достаточно гибким, поскольку малейшее изменение в архитектуре программно-аппаратного комплекса или в требованиях к системе приведет к необходимости к регенерации исходного кода и базы данных. Таким образом, при реализации любого из подходов возможности адаптации разработанного приложения сильно ограничены.

Специфика современного программного обеспечения такова, что жизненный цикл приложения никогда не заканчивается завершением его разработки, поэтому требуется обеспечить возможность динамической настройки системы на новые условия эксплуатации, потребности ее пользователей и автоматизируемых процессов.

Для создания адаптируемых систем мониторинга энергопотребления требуется хранить описание конфигурации системы (подключенных устройств, установленных параметров, режимов их отслеживания) и других настроек. При изменении конфигурации код приложения меняться не должен. Максимальная гибкость, таким образом, достигается, если работа программы строится на интерпретации этих описаний (моделей): по ним генерируется интерфейс пользователя и функционирует система, её программные компоненты. Именно режим интерпретации моделей позволяет достичь необходимой гибкости и

баланса между скоростью разработки, универсальностью и возможностями адаптации приложения.

### 2. Архитектура и принципы функционирования системы ЭнергоWatcher

Одной из задач, на решение которых направлено создание программного комплекса **ЭнергоWatcher**, является задача реализации системы, которая обеспечивала бы максимальную степень гибкости в сочетании с простотой, доступностью для различных категорий пользователей.

Создаваемый комплекс основан на многоуровневом моделировании в сочетании с работой в режиме интерпретации построенных моделей. При создании системы создается модель, включающая описание всех объектов, параметры которых должны собираться и обрабатываться.

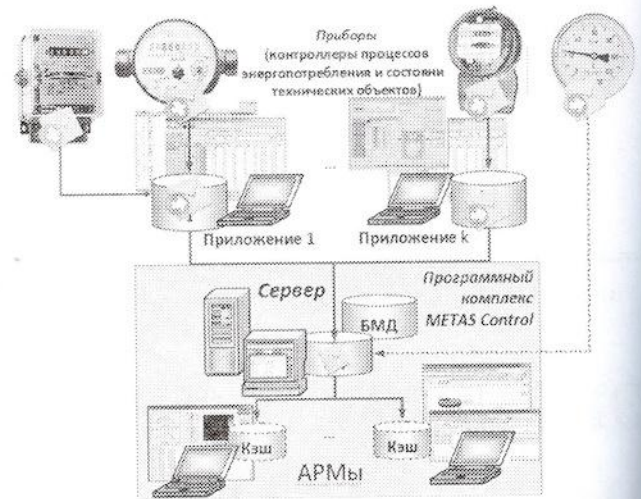


Рис. 1. Структура программно-аппаратного комплекса

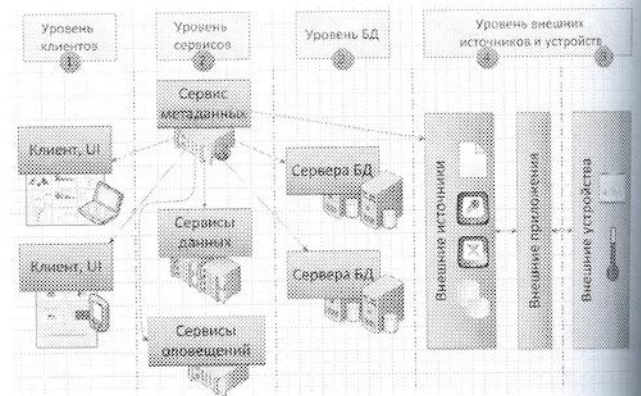


Рис. 2. Многоуровневая архитектура системы мониторинга ЭнергоWatcher

Ядро системы составляют взаимосвязанные метаданные, описывающие части системы с разных сторон. Модель метаданных (метамета-модель) логического уровня основана на ER-модели Чена (модель сущность-связь). Каждая сущность – это абстрактный объект реального мира (например, прибор учета), обладающая собственными свойствами (атрибутами) и отношениями с другими сущностями (связями). Экземпляры сущностей представ-

ляют существующие конкретные объекты, а их значения атрибутов и связей – их текущее состояние. При подключении нового прибора создается объект, соответствующий типу прибора, представляющий его в базе данных (БД) системы. Для каждого такого объекта определяется набор параметров, которые должны быть получены, и устанавливается связь с соответствующим источником (рис. 1). Данные могут быть получены как непосредственно с приборов, так и из баз данных приложений, поставляемых с приборами [1, 2]. Вся информация о системе хранится в базе метаданных (БМД).

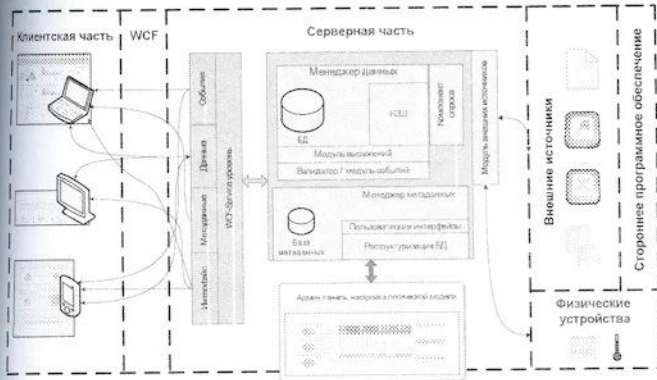


Рис. 3. Сервис-ориентированная архитектура ЭнергоWatcher

Система состоит из пяти (1-5) уровней. Уровни 4 и 5 отвечают за подключение к внешним источникам, настройку параметров импорта данных. Данные, получаемые с приборов учета потребления энергоресурсов, обрабатываются сторонними приложениями. После обработки полученные показатели приборов сохраняются во внешних источниках (базы данных программного обеспечения отдельных приборов, электронные таблицы, CSV-файлы и др.). Система опрашивает источники на наличие новых данных и при их наличии производит импорт в БД (уровень 3). Для каждой конфигурации оборудования на сервере БД (уровень 3) на основе метаданных создается собственная база данных для хранения выбранных для импорта показателей. Структура БД определяется метаданными логического и физического уровня согласно имеющейся конфигурации оборудования и обновляется автоматически при изменении этих метаданных с помощью компонента реструктуризации. Уровни 1 и 2 представляют собой части клиент-серверной сервис-ориентированной архитектуры (рис. 3). Конечный клиент устанавливает соединение с сервисами в режиме опроса/дуплекса и получает данные (сервисы данных и событий) и метаданные (сервисы метаданных и настроек UI). Сервис метаданных по запросу клиента предоставляет метаданные логического уровня, сервис UI – презентационного. Сервис событий автоматически оповещает подключенных клиентов при возникновении внештатных ситуаций, сервис данных – при поступлении новых импортированных значений. Клиент, в свою очередь, сам определяет собственный интерфейс панелей индикаторов (dashboard) и настраивает связь UI элементов управления с источниками данных.

Логическая связь данных уровней (рис. 4) обеспечивается взаимосвязанными слоями метаданных (МД), описывающих систему с разных точек зрения [3].

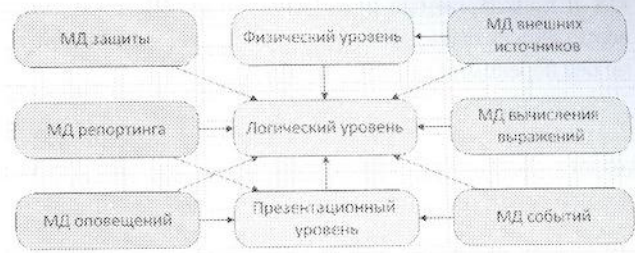


Рис. 4. Метаданные системы

Ключевая метамодель – модель логического уровня – обеспечивает взаимосвязь остальных метаданных и работу с системой в рамках терминах специализированной для пользователя предметной области (в данном случае в области энергоаудита).

### 3. Мультиязычный редактор UI и связывание с метаданными, конструктор WYSIWYG

Одна из главных целей создания приложений мониторинга – удобное и своевременное оповещение о происходящих процессах, внештатных ситуаций, отображение реальной картины потребления энергоресурсов (энергоаудита) для последующего анализа. Все это – задачи визуализации данных, которые в операционных панелях индикаторов имеют наиболее важное значение.

В общем случае для создания собственного интерфейса пользователю (или администратору) системы потребуется выполнить несколько действий:

- создать экземпляры компонентов визуализации (элементов управления);
- расположить данные элементы управления в нужных местах, учитывая их взаимную вложенность и иерархичность;
- настроить основные свойства элементов управления;
- указать источники данных для отображения – те атрибуты, значения которых требуется отобразить, и, при необходимости, задать недостающие значения.

Для обеспечения максимальной гибкости и адаптивности системы были определены следующие требования:

- настройка визуального интерфейса должна быть простой и понятной конечному пользователю-непрограммисту;
- набор поддерживаемых компонентов должен быть динамически расширяем без внесения изменений в код приложения;
- компоненты визуального интерфейса должны поддерживать один из возможных видов привязки данных (binding);
- все изменения в системе должны применяться «на лету», без перекомпиляции исходного кода, по принципу WYSIWYG (во многих системах модель интерфейса не интерпретируется во время выпол-

нения приложения, а лишь служит для генерации кода пользовательского интерфейса) [7].

Стоит отметить, что сами элементы управления, равно как и подключаемые источники данных должны иметь возможность быть динамически подключенными к системе без перекомпиляции исходного кода.

В технологии EnergoWatcher, основанной на интерпретации взаимосвязанных метаданных, пользователь создает собственную панель индикаторов, описываемую метаданными презентационного уровня. Пользователь сам выбирает, какие показатели и в какой конфигурации он хочет наблюдать, используя средства создания визуального интерфейса настраиваемой панели в EnergoWatcher,

а также предметно-ориентированные языки для задания ограничений, правил обработки данных.

### Заключение

Представленная технология позволяет создавать гибкие масштабируемые распределенные приложения для мониторинга параметров энергопотребления, которые могут работать параллельно с программами, поставляемыми производителями оборудования, взаимодействуя с ними и дополняя их.

Проект выполняется при поддержке РФФИ (проект № 12-07-00763-а).

### Список литературы:

1. **Воронов В.А., Калашников Е.А., Лядова Л.Н.** Технология создания системы мониторинга энергопотребления // Труды Конгресса по интеллектуальным системам и информационным технологиям «AIS-IT'10» Научное издание в 4-х томах. Т. 1. – М.: Физматлит, 2010. С. 493-500.
2. **Kalashnikov E.** EnergoWatcher – The Platform for Creating Adaptable Energy Monitoring Systems // Proceedings of the 6th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2012), May 30-31, 2012 – Perm, Russia. P.220-223.
3. **Лядова Л.Н.** Технология создания динамически адаптируемых информационных систем // Труды междунар. науч.-техн. конф. «Интеллектуальные системы» (AIS'07). Т. 2. – М.: Физматлит, 2007.