

А. В. Кычкин, канд. техн. наук, Национальный исследовательский университет «Высшая школа экономики», филиал в г. Перми, avkychkin@hse.ru

А. И. Дерябин, канд. техн. наук, доцент, Национальный исследовательский университет «Высшая школа экономики», филиал в г. Перми, aderyabin@hse.ru

О. Л. Викентьева, канд. техн. наук, доцент, Национальный исследовательский университет «Высшая школа экономики», филиал в г. Перми, ovikenteva@hse.ru

Л. В. Шестакова, канд. техн. наук, доцент, Национальный исследовательский университет «Высшая школа экономики», филиал в г. Перми, lshestakova@hse.ru

Проектирование IoT-платформы для управления энергоресурсами интеллектуальных зданий

Авторами рассматривается задача проектирования киберфизической системы, применяемой в качестве сервиса для управления интеллектуальными зданиями с использованием технологий Интернета вещей — Internet of Things (IoT). Такие программные платформы входят в состав комплексных систем класса BEMS — Building Energy Management System и являются инструментальным средством для реализации энергоресурсосберегающих мероприятий в зданиях. Сервера и контроллеры IoT образуют инфраструктуру системы управления, в которой особую роль играют механизмы адаптации и интеллектуального анализа данных, поступающих непрерывно с большого числа узлов. В условиях отсутствия стандартов по реализации BEMS систем на базе IoT, а также возрастания требований, предъявляемых к составу и функциям аналитического обеспечения, возникает необходимость в унификации программных архитектур и обеспечении их эффективной реализации при решении практических задач.

Ключевые слова: киберфизические системы, Интернет вещей, интеллектуальное здание, интеллектуальный анализ данных, облачные технологии, беспроводная передача данных, туманные вычисления, проактивное управление.

Введение

Среди программных систем мониторинга и управления энергоресурсами EMS (Energy Management System) наиболее популярны многофункциональные информационно-аналитические платформы таких производителей, как Oracle, Microsoft, IBM, Siemens, Schneider Electric, Centraline, АО «Группа ЭНТЕР», ГК «АйТи», НТЦ «АРГО», ЗАО «НПФ Прорыв» и других. Применение этих решений значительно повышает эффективность работы предприя-

тий за счет анализа энергетических данных и оперативного принятия решений в рамках системного энергетического менеджмента в аспекте различных видов ресурсов.

Несмотря на стремительный рост рынка EMS-систем, большинство доступных на рынке программных инструментальных средств для реализации веб-сервисов для управления энергоресурсами зданий остаются слишком сложными и дорогими, чтобы их правильно использовать. Их закрытая архитектура, невозможность адаптации аналитических компонент к конкретным сооруже-

ниям или условиям эксплуатации, высокие требования к квалификации энергоменеджеров не позволяют создавать эффективные объекты городской среды в условиях применения технологий Интернета вещей — Internet of Things (IoT). Наблюдается противоречие между уровнем развития технологий системного энергоменеджмента и формами реализации в широкой практике энергоресурсосберегающих мероприятий в зданиях. В связи с этим перспективным направлением исследований является построение открытых архитектур программных платформ для систем управления энергоресурсами зданий класса BEMS — Building Energy Management System, включающих средства обработки многомерных энергоданных с позиции энергоменеджмента.

Вопросы, связанные с использованием технологий IoT в управлении различными объектами, рассматриваются в работах [3–8] и др. IoT может помочь преодолеть проблему изолированных систем зданий [14] и поддержать создание единой среды. В этом новом контексте системы автоматизации зданий потенциально потребуют фундаментальных изменений в том, как они разрабатываются и устанавливаются.

В работах [6–7] обсуждаются различные технологии и проблемы, связанные с зеленым IoT, который дополнительно снижает потребление энергии. В работе [7] рассматривается облако датчиков, которое предполагает их использование в качестве интерфейса между физическим и цифровым миром. Облако датчиков (Sensor Cloud) [8] позволяет использовать их в ИТ-инфраструктуре путем виртуализации физического датчика на облачных вычислениях, т. е. представлять виртуальные датчики как услугу (SNaaS). Пользователям не нужно беспокоиться о местоположении физических датчиков, вместо этого они могут контролировать виртуальные датчики. Интеграция датчиков с облаком позволяет создать открытую, расширяемую, масштабируемую среду. В кон-

тексте SNaaS представляют интерес исследования [7], связанные с построением энергоэффективной архитектуры системы; стратегией использования энергоэффективных услуг; ситуационной и контекстной осведомленностью пользователей и приложений, например, обучение и прогнозирование; энергоэффективным управлением WSN (Wireless Sensor Networks); энергоэффективным управлением облаками.

В ряде работ [1, 2, 9–12] предлагается использовать облачные и туманные технологии при реализации программных систем для управления объектами. В работе [1] рассматривается возможность использования модели туманных вычислений как универсальной технологической платформы для проектов с применением Интернета вещей (IoT), мобильной связи 5G, искусственного интеллекта. В большей степени туманные вычисления подходят для работы с системами межмашинного взаимодействия и устройствами для сбора данных, генерируемых серийно расставленными сенсорами с частотой до долей секунд. Подобные устройства встречаются в системах безопасности и комплексах управления промышленными объектами. Эталонная архитектура туманных вычислений Openfog была опубликована в 2015 г. Особенностью туманных вычислений является локальная обработка собранных данных непосредственно на месте их генерации, что является определяющим фактором для систем реального времени. Данные размещаются на клиентских узлах, где предстоит их последующая обработка. Для сбора данных зачастую используется беспроводная связь.

В работе [2] предлагается технология конвергентных распределенных вычислений для сбора и обработки больших сенсорных данных, основанная на интеграции моделей GRID, облачных и туманных вычислений. Модель туманных вычислений используется для обработки и агрегирования сенсорных данных на уровне узлов сенсор-

ной сети и промышленных контроллеров. Для анализа агрегатов данных в центральном вычислительном кластере используются модели GRID и «облачных» вычислений.

В работе [8] рассматривается среда для разработки IoT-приложений, позволяющая использовать возможности облачных вычислений. IoT-приложение должно обеспечить:

- поддержку чтения потоков данных либо от сенсорных датчиков напрямую, либо из базы данных;
- реализацию функций обработки (анализа) данных в облачной инфраструктуре;
- визуализацию событий, обнаруженных в результате анализа данных.

В работе [9] предлагается архитектура, которая интегрирует IoT и аналитику больших данных. Данные IoT, собранные с различных датчиков и объектов, отличаются от обычных больших данных тем, что включают неоднородность, шум, разнообразие и быстрый рост. Статистические исследования [12] показывают, что к 2030 г. количество датчиков будет увеличено на 1 трлн. Это увеличение повлияет на рост больших данных. Авторы [9] выделяют сенсорный слой, содержащий сенсорные устройства и объекты, подключенные через беспроводную сеть. Беспроводная сетевая связь может быть реализована через RFID, Wi-Fi, ZigBee, Bluetooth и пр. Шлюз IoT позволяет осуществлять связь через Интернет и различные сети. Большой объем данных, полученных от датчиков, хранится в облаке. Данные доступны через приложения аналитики больших данных. Эти приложения содержат управление API и панель инструментов для обработки данных.

В работе [13] рассматривается программное обеспечение с открытым кодом — платформа VOLTTRON, предоставляющая услуги для управления различными активами в энергосистеме. Платформа разработана в Тихоокеанской северо-западной национальной лаборатории (США) и функционирует на основе программных агентов, которые могут принимать решения на основе информа-

ции, предоставленной оператором, другими агентами или онлайн-источниками в режиме реального времени. VOLTTRON-агенты делятся на три класса: агенты платформы служат поставщиками услуг на платформе; облачные агенты служат мостами между платформой и другими удаленными платформами; контрольные агенты управляют фактическими аппаратными устройствами.

Типовое архитектурное решение системы управления энергоресурсами здания

В качестве типового решения для реализации системы управления энергоресурсами здания выступает архитектура смешанного типа, включающая в себя архитектурные стили: вызов с возвратом (клиент-сервер), брокер сообщений и шина (рис. 1). Доступ к системе реализуется как с помощью настольного приложения, так и из веб-браузера и мобильного приложения [15]. К системе может быть подключено большое количество устройств, для этого в составе сервера реального времени может быть использован распределенный программный брокер сообщений Apache Kafka.

Сервера мобильных клиентов, настольного приложения, веб-клиента взаимодействуют с соответствующими им клиентами с помощью REST API. Сервер визуализации метрик Grafana доступен пользователю по HTTP-запросам.

Уровень клиента отвечает за взаимодействие системы с пользователями, графическое представление данных о потреблении ресурсов в реальном времени и статистической информации. На этом уровне осуществляется сбор данных с датчиков и формирование команд управления инженерными подсистемами зданий, включая подсистемы теплоснабжения, электроснабжения, освещения, водоснабжения, вентиляции, кондиционирования и др., с помощью исполнительных механизмов [16].

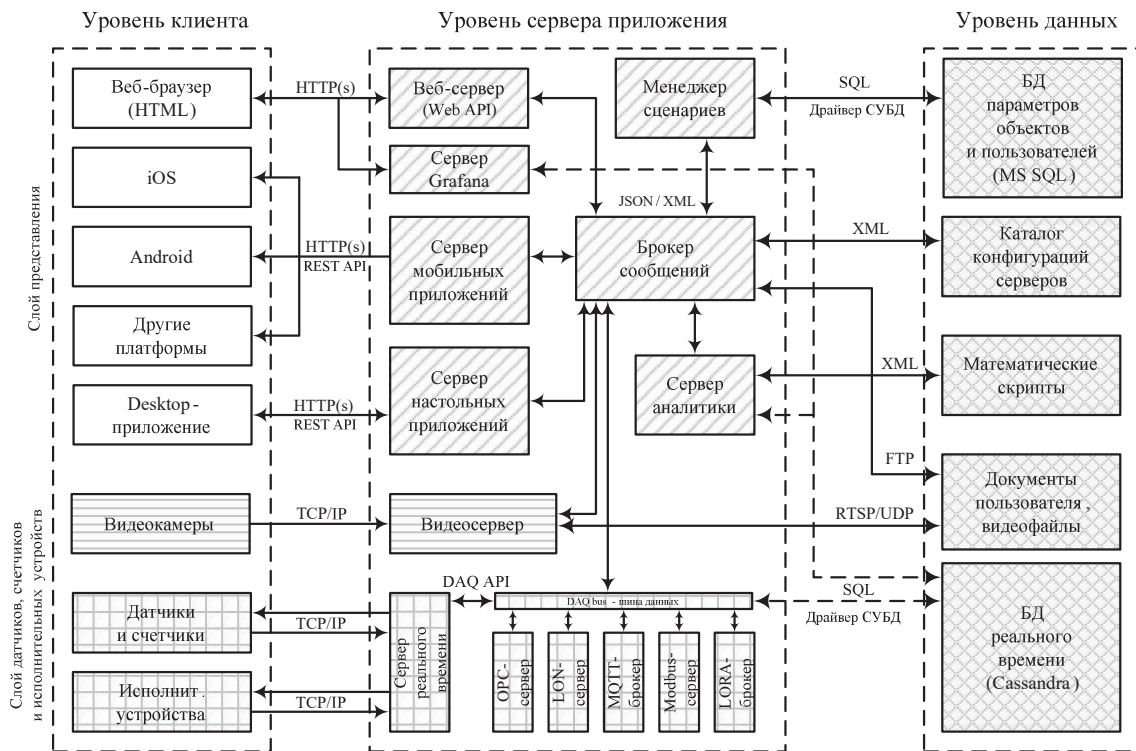


Рис. 1. Обобщенная архитектура системы управления энергоресурсами здания смешанного типа
 Fig. 1. The mixed type generalized building energy management system architecture

Уровень сервера приложений необходим для реализации функций управления, сбора и обработки информации как для последующего предоставления ее клиенту или исполнительным механизмам, так и для записи в базах данных. На данном уровне выделено шесть основных серверов: сервер мобильного приложения, сервер настольного приложения, сервер веб-приложения, сервер, обеспечивающий передачу данных в сервис Grafana, аналитический сервер, а также сервер, обрабатывающий данные с видеокamer, и сервер реального времени для связи с устройствами. Чтобы синхронизировать данные между серверами, используется брокер сообщений — отдельный модуль, занимающийся приемом и передачей сообщений между компонентами и модулями системы. Для обеспечения высокой скорости работы сервера реального времени с устройствами интеллектуального зда-

ния по различным протоколам целесообразно применение шины данных, доступ к которой реализуется посредством DAQAPI. Сервер аналитики выполняет оперативную обработку информации, анализ данных об энергопотреблении здания, распознавание аварийных ситуаций, динамическое построение отчетов в различных разрезах, прогнозирование событий с помощью OLAP-технологий, [17].

Уровень данных обеспечивает подключение к базам данных, а также прием и передачу информации от уровня сервера приложений. Уровень содержит СУБД MS SQL Server, предназначенную для хранения параметров системы, конфигураций серверов, информации о пользователе, а также видеоданных; кластеризуемую базу данных Cassandra или InfluxDB, предназначенные для хранения данных с датчиков в формате временных рядов.

Организация и принцип работы менеджера сценариев

Менеджер сценариев сервера приложений программной платформы осуществляет создание, редактирование, удаление, выбор и активацию сценариев управления объектами зданий, передает выполняемый в данный момент сценарий на сервер аналитики для моделирования его исполнения.

Зададим ограничения на подачу энерго-ресурсов j -го типа (электроэнергия, тепловая энергия, вода, пар, сжатый воздух, холод и др.) в здании для каждого момента времени на интервале T :

$$\forall t \in T : E_i^j \in [\underline{E}_i^j, \overline{E}_i^j], i = \overline{1, m}, j = \overline{1, n}$$

и общее ограничение на объем используемых энергоресурсов:

$$\sum_{i=1}^m \int_{t \in T_i} E_i^j(t) dt \leq E_0^j, \quad (1)$$

где $\underline{E}_i, \overline{E}_i$ — нижняя и верхняя границы подаваемой энергии или ресурса j -го типа в помещении i ; E_0^j — максимальный объем ресурса на временном интервале $T, T_i \subseteq T$ — временной интервал снабжения энергоресурсом i -го помещения.

Если $P = \{P_1, P_2, \dots, P_y; y = \overline{1, v}\}$ — сценарии, реализуемые алгоритмами работы программных компонентов, то оптимальный результат управления энергопотреблением здания будет достигнут при минимальных энергозатратах (1) по критерию:

$$K_j = \sum_{i=1}^m \int_{t \in T_i} E_i^j(t; P_y) dt \rightarrow \min(E_i^j), \quad (2)$$

где P — программно реализуемый сценарий управления инженерным оборудованием здания для i -го помещения. Выбор оптимального сценария может быть достигнут с помощью идентификации нелинейных зависимостей целевых значений климатических и других

показателей и объемов затрачиваемой энергии для заданных параметров, что реализуется в сервере аналитики с помощью механизмов интеллектуального анализа данных [18].

С программной точки зрения менеджер сценариев описывается классом «ScenarioManager». Атрибутом данного класса является коллекция экземпляров класса «Scenario», который описывает конкретный сценарий работы системы.

Ключевой метод — «setScenario()» по установленному набору сценариев управления объектами здания выбирает наиболее рациональный с точки зрения энергопотребления и активирует его исполнение на контроллере.

Это реализуется следующим образом. На первом этапе менеджер сценариев получает информацию о том, какой сценарий используется в данный момент на контроллере в исследуемом помещении здания.

Для заданных условий работы оборудования и помещения менеджер производит поиск в базе сценариев альтернативных вариантов, подходящих по классу управляемых переменных.

С помощью алгоритмов, реализуемых сервером аналитики, менеджер сценариев получает прогнозные значения по потреблению энергоресурсов в зависимости от анализируемого сценария и внешних условий.

Менеджер производит распознавание потенциальных аварийных ситуаций — делает оценку того, насколько прогнозируемая ситуация является аварийной в зависимости от сценария и последующих показаний датчиков.

Далее он решает задачу нахождения оптимального сценария работы оборудования на основе статистических и прогнозных данных по критерию K_j (2).

Формирует команду на активацию выбранного сценария для последующей отправки на контроллер управления с помощью следующих методов:

- «TaskSignalCreator ()» — формирование сигнала управления.

- «SignalSender ()» — отправка сигналов управления на контроллер.

Многоуровневая архитектура сервера аналитики уровня сервера приложений

Для реализации задач

$$Ts = \{Ts_1, Ts_2..Ts_a; a = \overline{1, n}\}$$

обработки и анализа временных рядов и числовых параметров об энергопотреблении здания на базе сервера аналитики предложены следующие архитектурные решения:

1. Компонентная архитектура, предусматривающая раздельное представление сервера аналитики с помощью нескольких компонентов $C = \{C_1, C_2..C_b; b = \overline{1, m}\}$, реализующих отдельные задачи.

2. Многоуровневая архитектура, где каждый вышестоящий уровень может обратиться только к нижестоящему.

3. Брокер сообщений на основе системы программных менеджеров

$$M = \{M_1, M_2..M_u; u = \overline{1, w}\},$$

реализующих управление вызовами функций и координирующих работу подсистем сервера аналитики $S = \{S_1, S_2..S_k; k = \overline{1, l}\}$.

В этом случае на верхнем первом уровне будет размещен менеджер анализа, координирующий работу подсистем S .

На втором уровне — компоненты, реализующие статистические расчеты и методы интеллектуального анализа данных по группам параметров.

На третьем уровне — менеджеры анализа временных рядов, содержащих информацию о потреблении энергии и ресурсов здания. Также этот уровень содержит компоненты подготовки данных для последующей обработки.

На четвертом уровне — обработчики временных рядов.

Обобщенную архитектуру A сервера аналитики программной платформы можно запи-

сать как операцию отображения целей и задач, а также состава менеджеров и подсистем в структурированное описание процесса обработки и анализа данных об энергопотреблении здания:

$$A : (Ts, M, S) \rightarrow P^C.$$

На рисунке 2 приведены основные компоненты и их взаимодействие друг с другом для программного блока, реализующего решение аналитических задач управления энергоресурсами зданий на уровне сервера приложений.

Представленная архитектура содержит следующие менеджеры:

- M_1 (класс «*Analysis manager*») — менеджер анализа, отвечающий за обработку системных и пользовательских вызовов функций обработки данных и служащий для координации работы подсистем S_1-S_7 . Входными данными являются запросы на выполнение какой-либо операции, а выходными — результаты их выполнения.

- M_2 (класс «*Time series manager*») — внутренний менеджер, управляющий подготовкой одного временного ряда для последующего анализа, включая выборку из БД и сохранение, предусматривающую поиск и исправление некорректных значений временных меток, устранение несистематических выбросов или провалов в измерениях, масштабирование и нормализацию.

- M_3 (класс «*Time series collection manager*») — внутренний менеджер, реализующий подготовку для последующего анализа нескольких временных рядов, включая масштабирование и калибровку значений по первому ряду, выравнивание временных меток, если они сделаны с разным шагом, фильтрацию и сглаживание.

- M_4 (класс «*Data manager for histogram presentation*») — внутренний менеджер, отвечающий за подготовку к представлению временного ряда в формате гистограмм. Формирует новый ряд данных, иллюстрирующий распределение случайной величины.

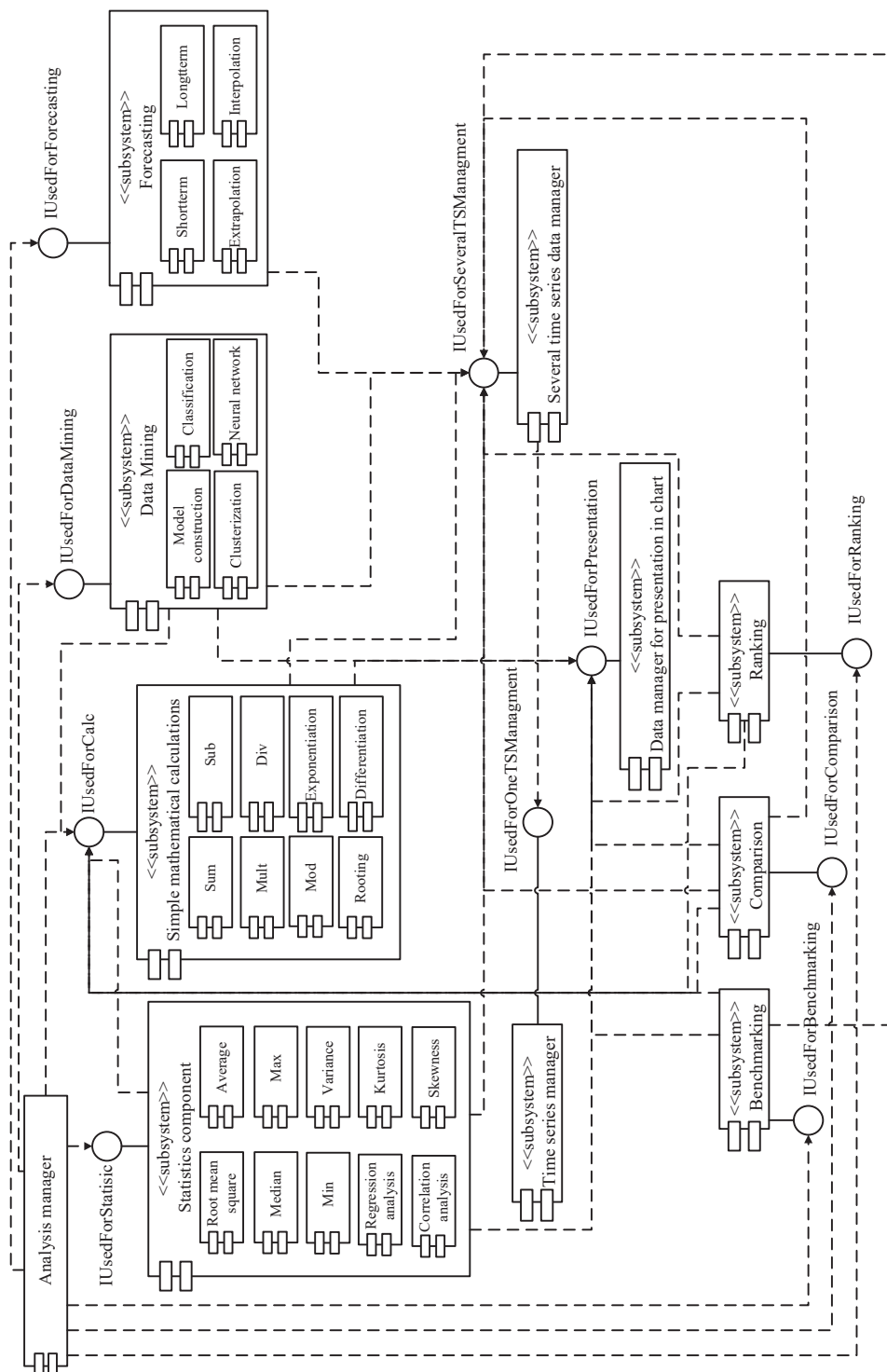


Рис. 2. Компонентная архитектура сервера аналитики программной платформы управления энергоресурсами здания
 Fig. 2. Component architecture of the BEMS software platform analytics server

Архитектура содержит следующие подсистемы:

- S_1 (класс «*Statistics*») — подсистема статистических расчетов, включая расчет среднего и среднеквадратичного значения энергопотребления, медианы, регрессионный анализ с построением нелинейных и линейных целевых функций энергопотребления, корреляционный анализ, используемый для оценки влияния различных факторов на энергопотребление здания.

- S_2 (класс «*BasicMath*») — подсистема реализации простейших математических операций над временными рядами данных.

- S_3 (класс «*Forecasting*») — подсистема прогнозирования, предназначенная для оценки тенденций и выявления закономерностей статистических показателей энергопотребления здания. Реализует функции краткосрочного прогнозирования (до 1 суток), долгосрочное прогнозирование (до 1 года), экстраполяции и интерполяции.

- S_4 (класс «*DataMining*») — подсистема интеллектуального анализа данных, предназначенная для выявления скрытых закономерностей и зависимостей в данных, а также идентификации модели исследуемых систем по результатам долгосрочных наблюдений на основе нейросетевых методов машинного обучения, кластеризации и классификации.

- S_5 (класс «*Comparison*») — подсистема количественной оценки отклонения нескольких временных рядов друг от друга в абсолютном и относительном эквиваленте по каждому измерению в масштабе эталонного (образцового) временного ряда.

- S_6 (класс «*Benchmarking*») — подсистема бенчмаркинга, реализующая упорядочивание временных рядов по значениям рассчитываемых интегральных показателей, например, площади, ограничиваемой профилем энергопотребления.

- S_7 (класс «*Ranking*») — подсистема ранжирования, осуществляющая поиск соотношения временного ряда к тому или иному рангу.

Реализация архитектуры программной системы управления энергоресурсами здания с помощью паттернов проектирования

Среди шаблонов проектирования выделяют порождающие, структурные и поведенческие. Несмотря на большое разнообразие паттернов, их применимость для решения поставленной задачи серьезно отличается.

Так, например, порождающий паттерн «Прототип» (*Prototype*) применим для создания экземпляров классов различных IoT-устройств, в том числе датчиков и счетчиков, обладающих схожей структурой и принципами работы в составе системы.

Порождающий паттерн «Строитель» (*Builder*) позволяет создавать сценарии в системе и регулировать доступ к функциям управления.

В качестве интерфейса высокого уровня для организации работы системы с сервером аналитики целесообразно применение структурного паттерна — «Фасад» (*Facade*). Фасад позволяет отделить компоненты сервера аналитики от системы и работать с ними независимо. В этом случае все внешние вызовы к сложно организованному аналитическим средствам сводятся к обращению к менеджеру анализов, распределяющему сообщения по подсистемам.

Поведенческий паттерн «Стратегия» (*Strategy*) позволяет выбирать один из нескольких созданных заранее сценариев управления по команде пользователя либо исходя из внешних условий, включая показания датчиков, время суток/года/день недели и т. д. С помощью данного шаблона создается гибкая архитектура классов сценариев, отвечающих за формирование нужных алгоритмов. Поскольку число компонентов, реализующих сценарии, в системе может быть велико, и каждый компонент может иметь несколько алгоритмов реали-

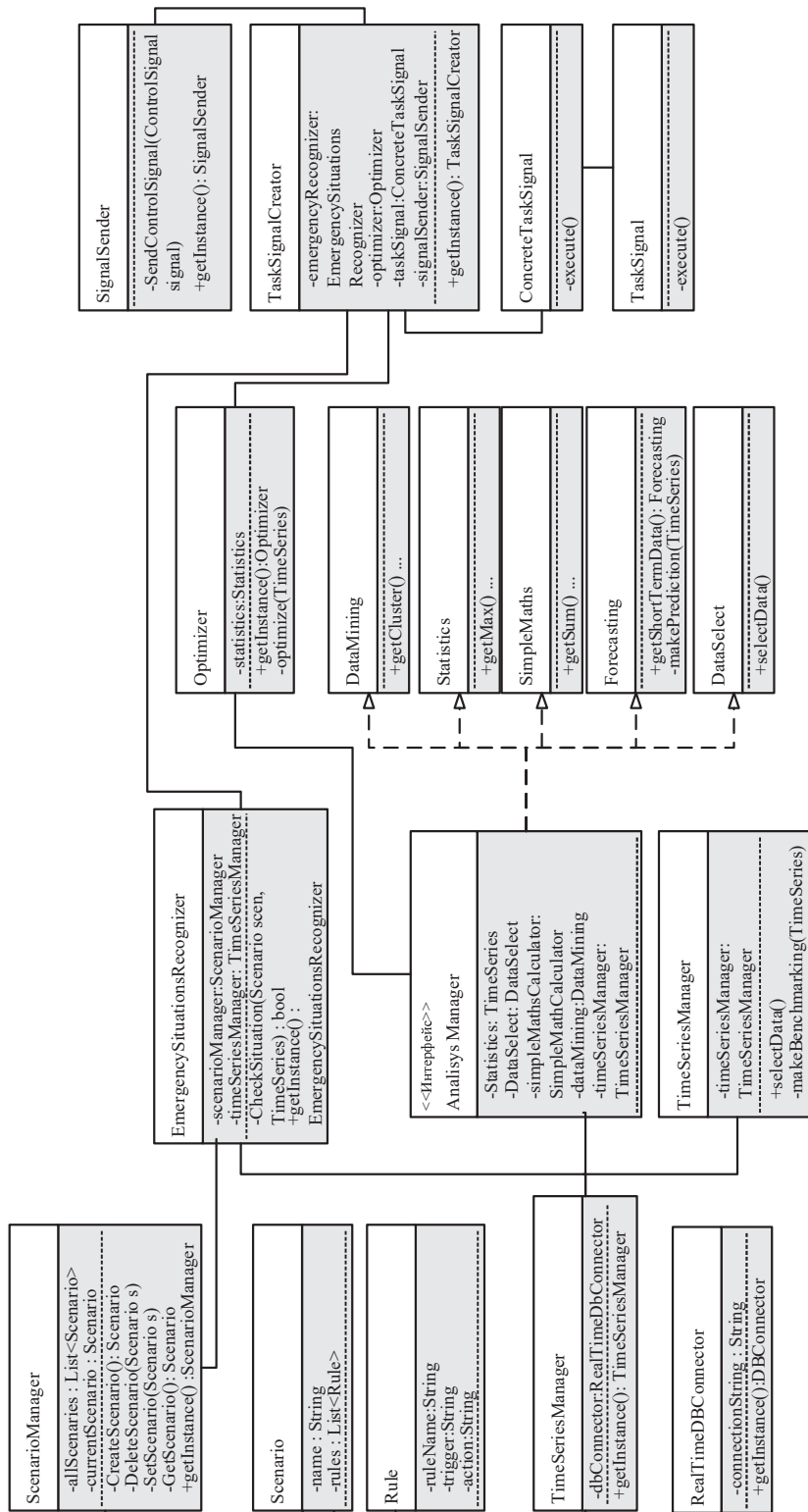


Рис. 3. Диаграмма классов программы сервера аналитики с применением поведенческих паттернов

Fig. 3. Class diagram of an analytics server program using behavioral patterns

зации, стратегия обеспечивает эффективные переключения между ними во время работы системы, в том числе с учетом возможных аварий и тревог.

Поведенческий паттерн «Команда» (*Command*) позволяет эффективно организовать выполнение последовательностей запросов к серверу аналитики, таких как расчет статистических показателей за определенный период времени, обработку и анализ данных, прогнозирование и т. д. Этот паттерн позволяет отменить затянувшийся процесс обработки данных и вернуть систему в исходное состояние.

Применение поведенческого паттерна «Посредник» (*Mediator*) снижает связность большого количества компонентов, которые работают совместно, что наблюдается в менеджере анализа класс — «*Analysis manager*». Данный класс содержит ссылки на все классы, содержащие алгоритмы обработки данных и отвечает за их последовательность выполнения (рис. 3).

Поведенческий паттерн «Цепочка обязанностей» (*Chain of responsibility*) подразумевает взаимодействие программных компонентов с целью обработки потока событий, запросов или сообщений. В данном случае

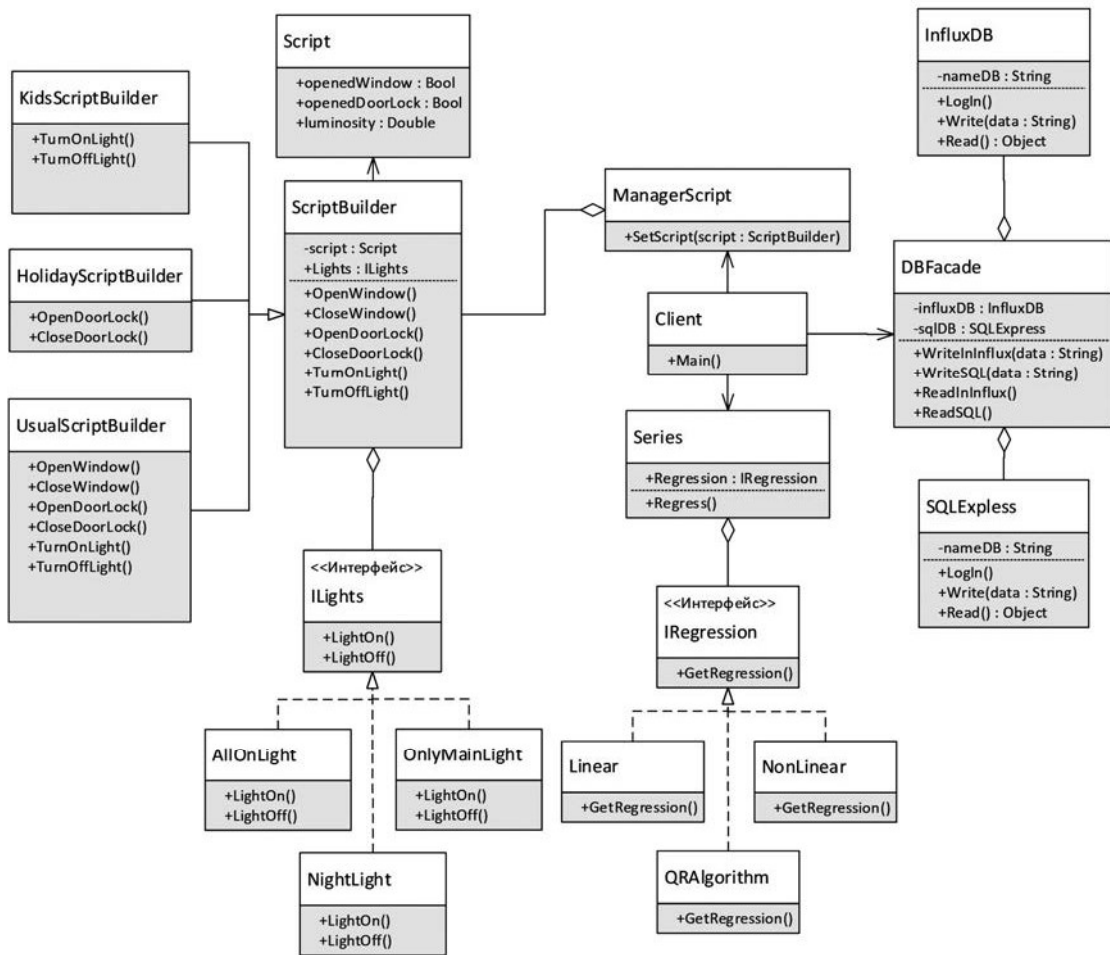


Рис. 4. Диаграмма классов составной части менеджера сценариев «ScenarioManager» с применением порождающего, структурного и поведенческого паттернов

Fig. 4. The ScenarioManager class diagram using the creational, structural and behavioral patterns

паттерн реализует последовательную обработку измерительной информации, передавая поток данных из одного компонента в другой, например, от менеджера M_2 подготовки временных рядов к подсистеме S_3 для формирования прогнозных значений. Применение «Цепочки обязанностей» позволяет менеджерам определять, куда дальше отправлять полученные в ходе обработки данные. Он наиболее эффективен в случаях, когда один запрос должен быть выполнен несколькими компонентами, порядок которых может определяться динамически.

На диаграмме классов (рис. 4) представлен состав одного из программных блоков менеджера сценариев (*ScenarioManager*), реализованного с помощью таких паттернов, как «Фасад», «Стратегия» и «Строитель».

В данном случае с помощью конкретных строителей *Kids Script Builder*, *Holiday Script Builder* и *Usual Script Builder* создаются объекты *ScriptBuilder* с определенным набором действий, определяющих сценарий управления зданием в зависимости от факторов человеческого присутствия. В роли распорядителя выступает класс менеджера *ManagerScript*, который вызывает только доступные методы конкретных строителей.

Заключение

Унификация организации и принципов работы базовых блоков проектируемой системы, включая менеджер сценариев, обеспечивает оперативную программную реализацию с использованием различных информационных технологий, в том числе облачных или туманных вычислений.

Представленные архитектурные шаблоны наглядно иллюстрируют возможности масштабирования и развития сервисов для управления интеллектуальными зданиями с использованием IoT.

Проектируемая программная платформа и комплекс реализуемых на ее основе сер-

висов являются в совокупности эффективным средством управления сложным энергетическим объектом или инженерной системой зданий с контуром интеллектуальной аналитики, построенной на основе открытой SaaS-архитектуры, способствующей развитию и тиражированию знаний об энергоэффективности зданий, включая использование возобновляемых источников энергии.

Список литературы

1. Новиков И. Консорциум OpenFog опубликовал эталонную архитектуру туманных вычислений. URL: <https://www.itweek.ru/infrastructure/article/detail.php?ID=192666>.
2. Финогеев А. А., Финогеев А. Г., Нефедова И. С. Технология конвергентной обработки данных в защищенной сети системы мониторинга // *Фундаментальные исследования*. 2015. № 11. С. 923–927.
3. Zanella A., Bui N., Vangelista L., Zorzi M. Internet of Things for Smart Cities // *IEEE Internet of Things Journal*. 2014. Vol. 1. No. 1. P. 22–32.
4. Thanh Nam Pham, Ming-Fong Tsai, Duc Binh Nguyen, Chyi-Ren Dow, Der-Jiunn Deng. A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies // *IEEE Access*. 2015. Vol. 3. P. 1581–1591.
5. Arshad Rushan, Saman Zahoor, Munam Ali Shah, Hongnian Yu. Green IoT: An Investigation on Energy Saving Practices for 2020 and Beyond // *IEEE Access*. URL: https://www.researchgate.net/publication/318797858_Green_IoT_An_Investigation_on_Energy_Saving_Practices_for_2020_and_Beyond?ev=publicSearchHeader&_sg=dWIS1SnLTai9MRQEJVbgBVlfx72XEbjgG2z1yHDC_ibUM354FF-CIOPg2dj_m43cLCxu86hqLzJWYkoc.
6. Chunsheng Zhu, Victor C. M. Leung, Lei Shu, Edith Ngai. Green Internet of Things for Smart World // *IEEE Access*. 2015. Vol. 3. P. 2151–2162.
7. Atif Alamri, Wasai shadab ansari, Mohammad Mehedi Hassan, M. Anwar Hossain. A Survey on Sensor-Cloud: Architecture, Applications, and Approaches // *International Journal of Distributed Sensor Networks*. 2013. URL: <http://dx.doi.org/10.1155/2013/917923>.
8. Кычкин А. В., Артемов С. А., Белоногов А. В. Распределенная система энергомониторинга реального времени на основе технологии IoT // *Датчики и системы*. 2017. № 8–9 (217). С. 49–55.
9. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions // *Future Generation Computer Systems*. 2013. Vol. 29. P. 1645–1660.

10. Mohsen Marjani, Fariza Nasaruddin, Abdullah Gani Mohsen. Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges // *IEEE Access*. Vol. 5. URL: <https://ieeexplore.ieee.org/document/7888916/>.
11. Chen M., Mao, S., Zhang, Y., Leung V. C. Big Data: Related Technologies, Challenges and Future Prospects. Springer Briefs in Computer Science, 2014. — 89 p.
12. Кычкин А. В. Программно-аппаратное обеспечение сетевого энергоучетного комплекса // Датчики и системы. 2016. № 7 (205). С. 24–32.
13. Katipamula S., Haack J., Hernandez G., Akyol B., Hagerman J. VOLTTRON: An Open-Source Software Platform of the Future // *IEEE Electrification Magazine*. 2016. № 4 (4). P. 15–22.
14. Stluka P., Parthasarathy G., Gabel S., Samad T. Architectures and Algorithms for Building Automation — An Industry View // *Intelligent Building Control Systems*. 2017. P. 11–43.
15. Викентьева О. Л., Дерябин А. И., Шестакова Л. В., Кычкин А. В. Синтез информационной системы управления подсистемами технического обеспечения интеллектуальных зданий // *Вестник МГСУ*. 2017. Т. 12. № 10 (109). С. 1191–1201.
16. Кычкин А. В. Синтез системы удаленного энергетического мониторинга производства // *Металлург*. 2015. № 9. С. 20–27.
17. Костызов А. М., Кычкин А. В., Борковец К. А. Прогнозирование электропотребления здания на основе селективного выбора нейронной сети // Информационно-измерительные и управляющие системы. 2017. Т. 15. № 9. С. 75–82.
18. Кычкин А. В., Микрюков Г. П. Метод обработки результатов мониторинга группы энергопотребителей // *Энергобезопасность и энергосбережение*. 2016. № 6. С. 9–14.
4. Thanh Nam Pham, Ming-Fong Tsai, Duc Binh Nguyen, Chyi-Ren Dow, Der-Jiunn Deng. A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies. *IEEE Access*, 2015, vol. 3, pp. 1581–1591.
5. Arshad Rushan, Saman Zahoor, Munam Ali Shah, Hongnian Yu. Green IoT: An Investigation on Energy Saving Practices for 2020 and Beyond. *IEEE Access*, Available at: https://www.researchgate.net/publication/318797858_Green_IoT_An_Investigation_on_Energy_Saving_Practices_for_2020_and_Beyond?ev=publicSearchHeader&_sg=dWIS1SnLTai9MRQEJVbgBVlfX72XEbjgG2z1yHDC_ibUM354FF-CIOPg2dj_m43cLCxu86hqLzJWYkoc (accessed 06.06.2018).
6. Chunsheng Zhu, Victor C. M. Leung, Lei Shu, Edith Ngai. Green Internet of Things for Smart World. *IEEE Access*, 2015, vol. 3, pp. 2151–2162.
7. Atif Alamri, Wasai shadab ansari, Mohammad Mehedi Hassan, M. Anwar Hossain. A Survey on Sensor-Cloud: Architecture, Applications, and Approaches. *International Journal of Distributed Sensor Networks*, 2013. Available at: <http://dx.doi.org/10.1155/2013/917923> (accessed 06.06.2018).
8. Kichkin A. V., Artemov S. A., Belonogov A. V. *Raspredeleonnaya sistema energomonitoringa real'nogo vremeni na osnove tekhnologii IoT* [Distributed real-time energy monitoring system based on IoT technology]. *Datchiki i sistemy* — Sensors and systems, 2017, no. 8–9 (217), pp. 49–55.
9. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 2013, vol. 29, pp. 1645–1660.
10. Mohsen Marjani, Fariza Nasaruddin, Abdullah Gani Mohsen. Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges. *IEEE Access*. Vol. 5. Available at: <https://ieeexplore.ieee.org/document/7888916/> (accessed 06.06.2018).

References

1. Novikov I. *Konsortium OpenFog opublikoval etalonnyu arkhitekturu tumannykh vychisleniy* [The Open-Fog Consortium published a reference architecture for foggy computing]. Available at: <https://www.itweek.ru/infrastructure/article/detail.php?ID=192666> (accessed 06.06.2018).
2. Finogeev A. A., Finogeev A. G., Nefedova I. S. *Tekhnologiya konvergentnoy obrabotki dannykh v zashchishchennoy seti sistemy monitoring* [The technology of convergent data processing in the protected network of the monitoring system]. *Fundamental'nyye issledovaniya* — Fundamental Research, 2015, no. 11, pp. 923–927.
3. Zanella A., Bui N., Vangelista L., Zorzi M. Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 2014, vol. 1, no. 1, pp. 22–32.
11. Chen M., Mao, S., Zhang, Y., Leung V. C. Big Data: Related Technologies, Challenges and Future Prospects. Springer Briefs in Computer Science, 2014, 89 p.
12. Kichkin A. V. *Programmno-apparatnoye obespecheniye setevogo energouchetnogo kompleksa* [Software and hardware of the network energy-accounting complex]. *Datchiki i sistemy* — Sensors and systems, 2016, no. 7 (205), pp. 24–32.
13. Katipamula S., Haack J., Hernandez G., Akyol B., Hagerman J. VOLTTRON: An Open-Source Software Platform of the Future. *IEEE Electrification Magazine*, 2016, no. 4 (4), pp. 15–22.
14. Stluka P., Parthasarathy G., Gabel S., Samad T. Architectures and Algorithms for Building Automation — An Industry View. *Intelligent Building Control Systems*, 2017, pp. 11–43.

15. Vikentyeva O. L., Deryabin, A. I., Shestakova L. V., Kichkin A. V. *Sintez informatsionnoy sistemy upravleniya podsystemami tekhnicheskogo obespecheniya intellektual'nykh zdaniy* [Synthesis of the information management system for the subsystems of the technical support of intellectual buildings]. *Vestnik MGSU — MSSU Bulletin*, 2017, vol. 12, no. 10 (109), pp. 1191–1201.
16. Kichkin A. V. *Sintez sistemy udalennogo energeticheskogo monitoringa proizvodstva* [Synthesis of the system of remote energy production monitoring]. *Metallurg — Metallurgist*, 2015, no. 9, pp. 20–27.
17. Kostygov A. M., Kichkin A. V., Borkovets K. A. *Prognozirovaniye elektropotrebleniya zdaniya na osnove selektivnogo vybora neyronnoy seti* [Forecasting power consumption of a building on the basis of a selective choice of a neural network]. *Informatsionno-izmeritel'nyye i upravlyayushchiye sistemy — Information-measuring and control systems*, 2017, vol. 15, no. 9, pp. 75–82.
18. Kichkin A. V., Mikryukov G. P. *Metod obrabotki rezul'tatov monitoringa gruppy energopotrebiteley* [Method of processing the results of monitoring the group of energy consumers]. *Energobezopasnost' i energosberezheniye — Energy Safety and Energy Economy*, 2016, no. 6, pp. 9–14.

A. Kychkin, National Research University Higher School of Economics Campus in Perm, Perm, Russia, avkychkin@hse.ru

A. Deryabin, National Research University Higher School of Economics Campus in Perm, Perm, Russia, aderyabin@hse.ru

O. Vikentjeva, National Research University Higher School of Economics Campus in Perm, Perm, Russia, ovikenteva@hse.ru

L. Shestakova, National Research University Higher School of Economics Campus in Perm, Perm, Russia, lshestakova@hse.ru

IoT-platform design for smart buildings energy management

The problem of designing a cyberphysical system used as a service for Smart buildings control using Internet technologies — Internet of Things (IoT) is considered. Such software platforms are part of the complex systems of the BEMS — Building Energy Management Systems and are an instrument for implementing energy savings in buildings. IoT servers and controllers form a management infrastructure in which the mechanisms of adaptation and data mining, which are continuously received from a large number of nodes, play a special role. In the absence of standards for the implementation of BEMS based on IoT, as well as increasing requirements for the composition and functions of analytical support, there is a need to unify software architectures and ensure their effective implementation in solving practical problems. Software component unification, including the scenario manager, provides operational software implementation using various information technologies, including cloud or foggy computing. The presented architectural templates clearly illustrate the possibilities of scaling and developing services for managing intelligent buildings using IoT. The projected open SaaS architecture software platform is an effective means for buildings energy managing with a contour of Data Mining. It can be used for replication of knowledge about the energy efficiency in buildings, including the use of renewable energy sources.

Keywords: cyberphysical systems, Internet of Things, Smart Building, data mining, cloud computing, wireless data transmission, fog computing, proactive control.

About authors:

A. Kychkin, *PhD in Technique*;

A. Deryabin, *PhD in Technique, Associate Professor*;

O. Vikentjeva, *PhD in Technique, Associate Professor*;

L. Shestakova, *PhD in Physics & Mathematics, Associate Professor*

For citation:

Kychkin A., Deryabin A., Vikentjeva O., Shestakova L. IoT-platform design for smart buildings energy management. *Prikladnaya informatika — Journal of Applied Informatics*, 2018, vol. 13, no. 4 (76), pp. 29–41 (in Russian).