

Применение ТРИЗ при проектировании архитектуры системы управления на основе данных в энергетике

Тюков Антон Павлович, Волгоградский государственный технический университет,
пр. Ленина 28, Волгоград, Россия

Аннотация

В статье описан опыт визуализации архитектуры системы управления на основе данных в энергетике (СУНД) и формирование стратегии ее развития с использованием инструментов ТРИЗ: функциональный анализ, причинно-следственный анализ СЕСА, поиск недостатков и приемов устранения недостатков. СУНД предназначена для управления информационными потоками в энергетике: управление отоплением с использованием прогноза погоды, управление зарядкой электромобилей, управление энергетическими потоками в распределенных энергетических системах, технико-экономическое обоснование. Анализ и формирование новой архитектуры системы производится с помощью нотации UML, BPMN, Archimate, DFD в среде моделирования систем Visual Paradigm с использованием методологии TOGAF. Автор высказывает гипотезу наличия собственных обобщенных недостатков и приемов их устранения при формировании архитектур систем управления на основе данных.

Ключевые слова: информационные системы, архитектуры, ТРИЗ, управление на основе данных, системная инженерия

Abstract

The author outlines the practical experience of describing the architecture of a data-based energy management system (SDMS) and the formation of a development strategy using TRIZ tools: functional analysis, causal analysis, the formation of disadvantages, the use of methods for eliminating disadvantages. SUND is designed to execute algorithms in real time to control energy flows in the energy sector: heating control through weather forecast, electric vehicle charging control, energy flow control in distributed energy systems. Analysis and formation of a new system architecture is performed using the notation UML, BPMN, Archimate, DFD in the Visual Paradigm system modeling environment using the TOGAF methodology. The author hypothesizes that there are heuristics for eliminating disadvantages in the formation of data-based control systems architectures.

Keywords: минимум три слова, отражающих тематику вашей статьи на английском языке.

1. Введение

Системы управления на основе данных относятся к категориям сложных систем, так как один человек не сможет удержать описание использованием доступных инструментов[1].

В данной статье рассматривается архитектура системы управления на основе данных в области энергетики, разрабатываемая в течение 10 лет 50+ разработчиками: менялись потребители продуктов, разработчики, программы финансирования, технологии, но не описана архитектура системы и нет стандартов. Компания ускорила свое развитие, что начинает формировать дополнительные требования к архитектуре системы управления на основе данных.

Цель статьи - рассказать об опыте описания и развитии системы управления на

основе данных за счет использования ТРИЗ, состоит из следующих разделов:

- 1) Анализ задачи проектирования архитектуры системы управления на основе данных.
- 2) Описание решения: рассказывает об основной идее предлагаемого решения.
- 3) Обсуждения и выводы: содержит предложенный список недостатков и приемов их устранения.
- 4) Результаты внедрения.

Авторы считают, что использование ТРИЗ позволяет повысить эффективность управления развитием системы управления на основе данных, предположили гипотезы о наличии эвристик при проектировании информационных систем, схожих с 30 недостатками, предложенными В. Леяшиным и 40 приемами устранения противоречий в “железной” ТРИЗ [2],[3],[4].

2. Анализ задачи

Сотрудники компании до внедрения описанных методов использовали разнообразные разрозненные инструменты проектирования, создания, документирования систем управления на основе данных, технологические требования не описывались, что приводило к следующим недостаткам:

1) Разработчики обладают узким кругозором относительно всей системы и глубокими знаниями о своей, что приводит к постоянным спорам относительно архитектуры системы, используемых компонентах, дублированию функционала и частого переопределения и переписывания системы.

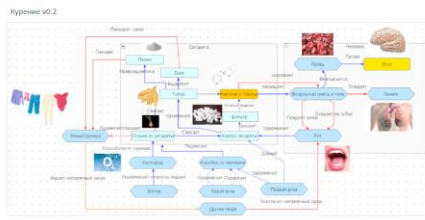
2) Отсутствие единого глоссария негативно влияло на процесс проектирования архитектуры решения, возникали вопросы с определением такого слова, как продукт, сервис, соглашение, соглашение на уровень сервиса и т.д.

3) Используется некоторое количество программ визуализации архитектуры: Draw.io, Visio, Flying Logic, что не позволяет повторно использовать диаграммы других проектах и обмениваться проектами между участниками.

4) Отсутствие описанной технической системы не дает пройти due diligence в крупных проектах.

Перед началом работы с архитектурой приложения определены следующие требования: i) программа должна моделировать систему, а не рисовать диаграммы, ii) работать с уровнями описания технических систем iii), повторно использовать описанные объекты, iv) хранить все разработанные диаграммы в одном файле, v) поддерживать репозиторий и совместное хранение и редактирование архитектуры онлайн. Разработчики выбрали Visual Paradigm как единственную программу, отвечающую поставленным требованиям.

В качестве прототипа процесса проектирования архитектуры использовался процесс обучения Мастера ТРИЗ Юрия Даниловского по функциональному анализу, состоящий из следующих этапов: 1) Проведение функционального анализа в нотации Майлза, 2) Определение списка недостатков, 3) Проведение причинно-следственного анализа первопричин, проектирование списка недостающих компонент.

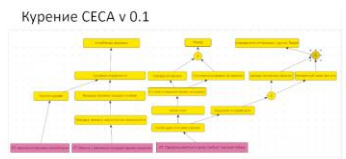


Функциональный анализ

Список недостатков

1. Сигарета создает опасное поле. Недостаток 1. Огонек сигареты может испортить одежду.
2. Сигарета создает опасное поле. Недостаток 2. Дым сигареты пропитывает одежду, она начинает издавать неприятных запах
3. Никотин добавляет вещество на зубы. Недостаток 3. Зубы меняют окраску.
4. Дым от сигареты создает опасное поле. Недостаток 4. У курильщика появляется неприятный запах изо рта, который чувствуют другие люди.
5. Дым от сигареты трансформирует вещество. Недостаток 5. Вредные вещества оседают в легких.

Формулизация недостатков



Проведение SECA

Рисунок 4 - Проведение функционального анализа на учебном примере Сигарета в авторском исполнении Антона Тюкова.

В описанном решении вместо диаграмм функционального анализа используются диаграммы потоков данных (DFD) как инструмент анализа информационных потоков. В работе использовалось понятие недостаток, вместо привычного в ТРИЗ противоречия из-за того, что противоречия сложны для восприятия неподготовленными людьми.

3. Описание решения

Предлагаемая методика позволяет структурировать процесс управления развитием управления на основе данных, снизить дублирование компонент, снизить транзакционные издержки при коммуникации кросс-дисциплинарных географически распределенных команд и основана на:

- 1) Использовании стандарта описания архитектуры организации TOGAF для определения аспектов внимания при проектировании архитектуры предприятия.
- 2) Визуализации отдельных аспектов системы на основе применения общепринятых стандартов визуального проектирования систем управления на основе данных:
 - a) data flow diagram (DFD, аналог функционального анализа для информационных систем);
 - b) BPMN для описания процессов, Archimate отображения архитектуры организации.
 - c) Принципах функционального анализа технических систем в нотации Майлза перенесенных на архитектуры информационных систем и нотацию DFD.
 - d) Процесс поиска недостатков на основе нотации ТРИЗ и приемов их устранения.

Реализация методики управления развитием системы управления на основе данных состоит из следующих этапов:

- 1) Формирование списка программного обеспечения, написанного и запущенного в экосистеме управления на основе данных, определение списка людей, ответственных за каждое разработанное программное обеспечение.
- 2) Формирование графика встреч с каждым из разработчиков для обсуждения элементов архитектуры системы. Для формирования полного представления требуется несколько итераций проработки используемого функционала с коллегами.
- 3) Создание синтезирующих диаграмм, позволяющих обобщить и сгруппировать созданные архитектурные описания.

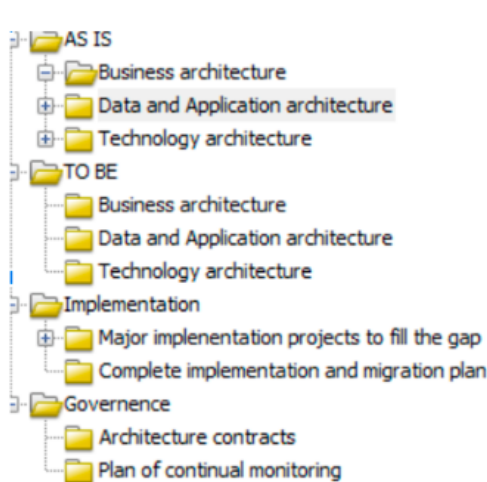
- 4) Создание полностью единой, полностью синтетической диаграммы, описывающей концепцию системы управления на основе данных, которая позволяет: i) обзорно описать информационную систему управления на основе данных в понятном для нетехнологического специалиста формате ii) создать маршрутизацию доступа для диаграмм.
- 5) Проведение анализа всей экосистемы систем управления на основе данных для выделения недостатков, проведение причинно-следственного анализа для перехода к состоянию “Как должно быть”.
- 6) Составление группы диаграмм “Как должно быть” идеального состояния системы, на основании анализа текущего состояния.
- 7) Создание онлайн репозитория для коллективной работы с диаграммами.
- 8) Публикация результатов построения диаграмм в виде html файла в единую экосистему пользователя для вывода информации в экосистеме компании.
- 9) Обучение сотрудников работать с системой.

После проведения итерации на основе анализа технических систем возможен переход к более обширному представлению архитектуры организации для синхронизации разработанного описания программной части архитектур с бизнес-процессами клиента и технической реализацией.

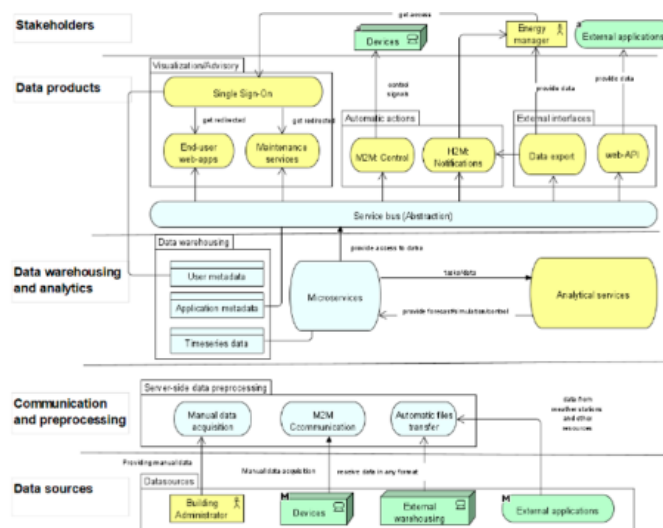
4. Результаты внедрения

В течение трехмесячного проекта было построено более 80 диаграмм, раскрывающих суть различных уровней системы в состоянии “Как есть”, “Как должно быть”, содержащие диаграммы, описывающие процесс преобразований и поддержания архитектуры системы. Уровни организации были раскрыты на основе рекомендаций методологии TOGAF: бизнес-уровень, уровень данных и приложений, уровень технологий.

В рамках бизнес-уровня описываются: 1) бизнес-процессы клиента при получении пользы от разрабатываемой системы управления на основе данных, 2) процесс продажи и обслуживания клиента, 3) описание бизнес-процесса разработки новых компонент информационной системы. В большинстве случаев, используется нотация BPMN.



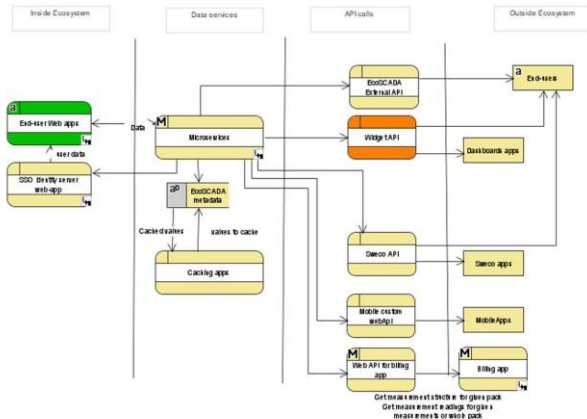
Логическая структура для группировки диаграмм



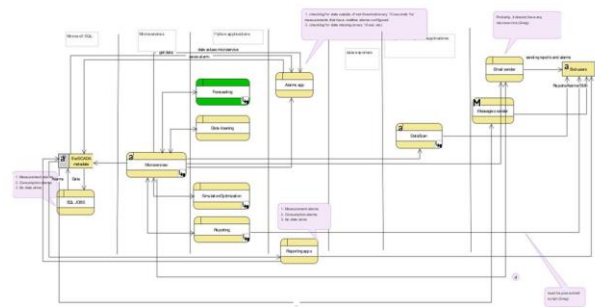
Маршрутизирующая диаграмма

Рисунок 1 - Логическая структура группировки диаграмм и маршрутизируемая диаграмма (1 уровень)

Данная архитектура логически разделена на следующие функциональные блоки: источники данных, предобработка данных с объединением, хранение данных и аналитика, сервисы на основе данных. Эта диаграмма используется только для анализа системности компонент и маршрутизации других диаграмм, составленных с участием разработчиков.

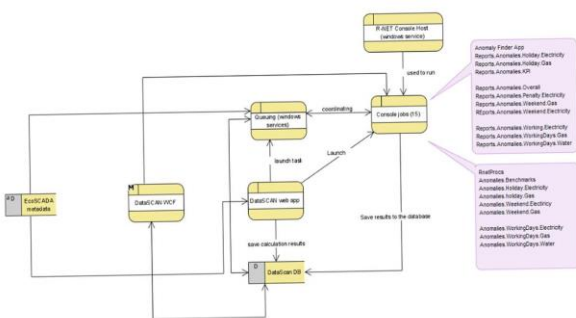


Хранение, кэширование и выгрузка данных



Управление алгоритмами управления на основе данных

Рисунок 2 - Логическое представление архитектуры данных в системе (2 уровень)



Автоматическая система создания отчетов на основе анализа данных Архитектура управления распределенной энергетической системой (3 уровень)

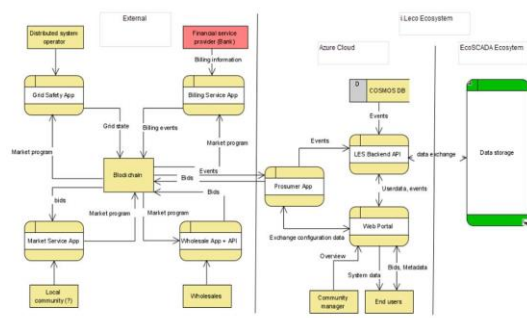
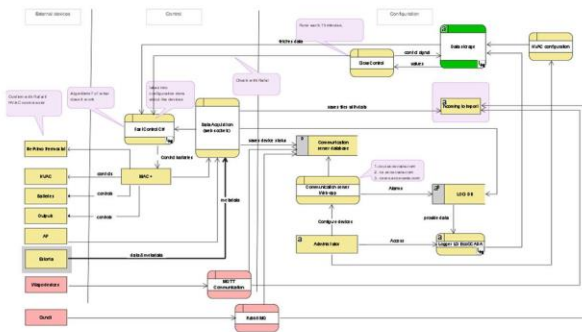
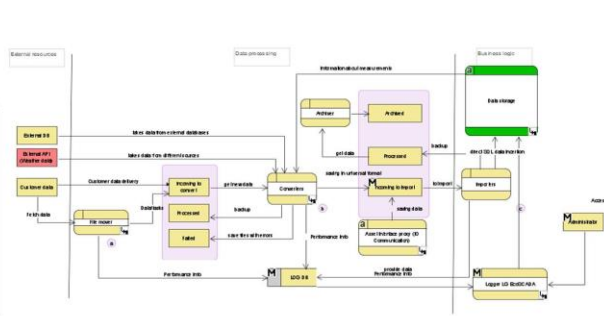


Рисунок 3 - Визуальное описание некоторых выбранных приложений (3 уровень)

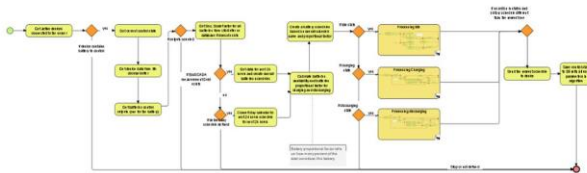


Процесс выгрузки и загрузки данных



Процессы управления в управлении распределенной системы управления на основе данных

Рисунок 4 - Процесс выгрузки, загрузки данных и управление распределенным оборудованием.



Описание алгоритмов управления батареями (4 уровень)

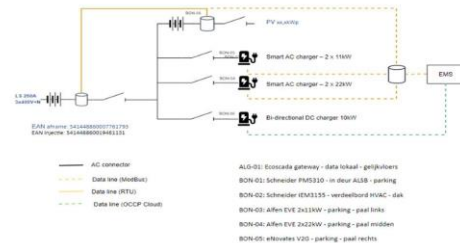


Диаграмма подключений при заправки электромобилей (4 уровень)

Рисунок 5 - Примеры описания алгоритмов управления и диаграммы подключения

5. Обсуждение и выводы

В рамках исследований разработан глоссарий системы, который позволяет прийти к единому языку описания системы, построена и визуализирована архитектура. Наличие диаграмм снизило трудозатраты при объяснении сложных задач, распределении труда разработчиков, формировании структуры отделов компании. В рамках анализа была предложена гипотеза о списке недостатков в архитектурах управления на основе данных. В качестве классификации недостатков были выбраны 7 альф системной инженерии[2]: i) стейкхолдеры, ii) возможности, iii) определение системы, iv) воплощение системы, v) команда, vi) работа, vii) технология. Информация с описанием предложенных недостатков и приемов их устранения представлена в Таблице 1.

Таблица 1 - Предложенные обобщенные недостатки и приемы устранения недостатков в ИТ

Наименование недостатка	Причина возникновения	Прием устранения
Низкая оперативность передачи данных (Воплощение системы)	Возникает из-за слишком большого количества компонент-посредников в системе, низкой пропускной способности информационного канала, особенности технологического оборудования	Убрать количество компонент-посредников, передать функции в надсистему, реализовать систему кэширования
Наличие посредников-компонент обработки данных (Воплощение системы)	Возникает, когда разработчики не синхронизируют свою работу	Согласовать деятельность разработчиков (общая архитектура системы)
Дублирование компонент (Воплощение системы)	Возникает, когда в экосистеме присутствует несколько компонент, делающих одно и то же	Согласовать с разработчиками, перестать поддерживать одну из компонент
Низкая согласованность технологий (Технологии)	Возникает тогда, когда один и тот же функционал реализован различными компонентами	Создание документа Architecture Definition в компании
Дублирование функций в различных приложениях (Определение системы)	Несколько приложений повторяет полностью один и тот же функционал	Перераспределение функций между компонентами, проектирование системы таким образом, чтобы она входила в одну функциональную группу
Низкая стандартизация программного обеспечения на уровне технологий (Технологии)	Приложения одного и того же типа разрабатываются на разных языках. Возникают проблемы с обучением, передачи кода и т.д.	Создание документа Architecture Definition в компании
Несогласованность функций в экосистеме (Определение системы)	Несколько приложений частично содержат схожие функции. За полным функционалом пользователю необходимо использовать несколько приложений	Перераспределение функций между компонентами в соответствии с функциональными группами
Недостаточная прозрачность при принятии решения (Стейкхолдеры)	Приложение показывает какую-то важную информацию, но пользователю непонятно, что с этим делать	Перейти от поддержки принятия решений к управлению, добавить в систему модуль пояснений.

Избыточные трудозатраты при получении информации (Стейкхолдеры)	Для своевременного получения информации требуется слишком много усилий	Организовать доставку информации по наиболее доступным для пользователя каналам: в мобильный телефон, на почту.
Низкая вовлеченность пользователя в работу приложения (Стейкхолдеры)	Возникает тогда, когда пользователи не понимают собственных целей работы в системе	Разработать систему мотиваций для отслеживания изменений в системе. Формировать контур обратной связи с помощью предоставления статистик и игрофикации.
Долгое время подготовки данных перед анализом (Команда)	Для подготовки данных требуется слишком много времени	Стандартизировать формат получаемых данных, создать систему автоматической предобработки данных перед анализом
Отсутствие прозрачности при контроле за техническими процессами (Команда)	Неизвестно, какие компоненты системы работают, а какие нет.	Создание связки клиент-сервер-данные-качество данных для постоянного отслеживания фактического достижения договоренностей с пользователем.
Избыточная связность между приложениями (Команда)	Приложения при своей работе вызывают друг друга без объективной необходимости, это приводит к снижению устойчивости системы	Провести рефакторинг, уменьшить связность между компонентами в системе
Недостаточная осведомленность руководителя относительно текущей архитектуры (Стейкхолдеры)	Происходит тогда, когда слишком высокая скорость изменений, низкое качество описания системы, отсутствует система проектирования и планирования	Сформировать стандарты описания компонент системы
Недостаточный уровень качества компонент (Воплощение системы)	Возникает запрос на поиск альтернативных решений при разработке программного обеспечения	Разработать инструмент оценки качества компонент
Низкая скорость заказа на изменения	Возникает в случае медлительности менеджмента при проектировании изменений, приводит к дублированию компонент	Увеличивать осведомленность людей, поддерживать стандарты качества, быстро реагировать на изменения.

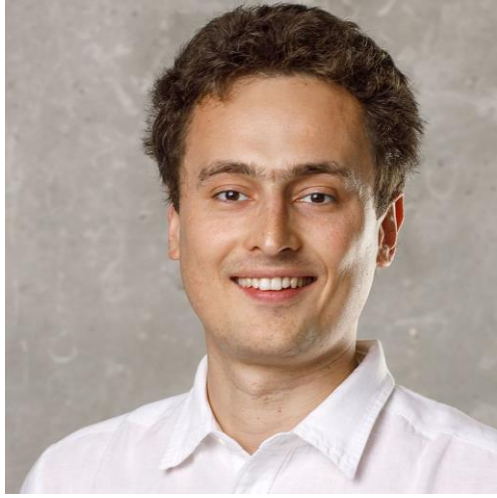
Данный способ классификации не является всеобъемлющим, но позволяет снять психологическую инерцию архитектора информационной системы. Настоящие рекомендации могут корректироваться и расширяться.

6. Заключение

В данной статье авторы предложили способ анализа и улучшения систем управления на основе данных с использованием ТРИЗ, сформулировали недостатки архитектур систем управления на основе данных и предложили приемы для их устранения. Авторами предполагается, что дальнейшая работа по структурированию недостатков и приемов позволит развить ТРИЗ для проектирования архитектур управления на основе данных, что в потенциале позволит увеличить аналитическую мощь архитекторов. Исследование может быть развито в следующих направлениях: i) Расширение описания системы на бизнес-процессы пользователя, объединение описаний бизнес-процессов пользователя с функциональным приложения, возможно использования статистик пользователей, ii) разработка онтологии процесса стейкхолдера, в рамках которого он получает ценность при использовании сервиса компании, iii) структурирование недостатков и приемов.

Список литературы

1. Левенчук А.А. Системное мышление 2019/ А.А. Левенчук - Издательские решения, Москва, 2019 - С. 340.
2. Альтшуллер Г. С., Верткин И. М. Как стать гением: жизненная стратегия творческой личности./ Г. С. Альтшуллер, И. М. Вёрткин. – Минск: Беларусь, 1994. – 479 с.
3. Классификация потребностей в практике ТРИЗ-консультирования/ Л. Х. Певзнер. // ТРИЗ в развитии/ Сборник научно-исследовательских трудов. Библиотека Саммита разработчиков ТРИЗ.- Санкт-Петербург, 2016. – Выпуск 8. – С. 268-280.
4. Рубин М.С. Основы ТРИЗ и инновации применение ТРИЗ в программных и информационных системах. / Рубин М.С., Кияев В.И.- Санкт. Петербург: Изд-во Санкт-Петербургского гос. ун-та, 2011, С 280.
5. Нил Форд,. 97 этюдов для архитекторов программных систем / Майкл Найгард, Билл Де Ора и Др. - Санкт-Петербург - Москва: Изд-во Символ, 2010, С 240.
6. **Автор для контакта**



Тюков Антон Павлович,
кандидат технических наук,
anton.tyukov@gmail.com