

РАЗРАБОТКА СИСТЕМЫ УВЕДОМЛЕНИЙ ОБ АВАРИЯХ НА ПРЕДПРИЯТИЯХ В ВИДЕ TELEGRAM-БОТА С ПОМОЩЬЮ ПЛАТФОРМЫ CAMUNDA

DEVELOPMENT OF A SYSTEM NOTIFYING ABOUT EMERGENCIES AT ENTERPRISES IN THE FORM OF A TELEGRAM BOT USING THE CAMUNDA PLATFORM

**Yu. Savelieva
N. Kulagina
V. Vaulin
A. Luk'yanov**

Summary. Most modern enterprises introduce into production and utilize advanced automation solutions at various stages of the product lifecycle. A wide range of software is used for management systems, ranging from tools for individual components of technological processes and equipment to complex software developed for managing technological processes, production, and entire enterprises. These systems also provide data management, including information transfer and delivery both within the company and externally, depending on the need. Some existing systems for continuous monitoring and control of industrial equipment allow data management not only through software itself, local networks, and the Internet, but also via mobile devices, such as SMS notifications. However, this method is not always convenient for the staff of the enterprise, who often prefer notifications through messengers or working chats that deliver specific messages with relevant information. To notify about accidents related to industrial equipment, a Telegram-bot was developed using the Camunda software platform. The algorithm of the system notifying about emergencies at enterprises presented in the work can be used in various areas, including non-production areas, to generate notifications.

Keywords: automation engineering solutions, industrial equipment, alert systems, notification systems, Camunda, Telegram bot.

Савельева Юлия Олеговна

канд. техн. наук, филиал ФГБОУ ВО

Самарского государственного технического

университета в г. Сызрани

Savelieva_yu_ol@mail.ru

Кулагина Надежда Алексеевна

филиал ФГБОУ ВО Самарского государственного

технического университета в г. Сызрани

Ваулин Владимир Иванович

канд. пед. наук, филиал ФГБОУ ВО

Самарского государственного технического

университета в г. Сызрани

Лукьянов Александр Александрович

филиал ФГБОУ ВО Самарского государственного

технического университета в г. Сызрани

Аннотация. Большинство современных предприятий внедряют и используют передовые решения в сфере автоматизации на отдельных или всех этапах жизненного цикла продукта. Применяется различное программное обеспечение для систем управления: от программных сред для отдельных узлов технологического процесса и оборудования до сложных программных комплексов для управления технологическими процессами, производствами и предприятиями, обеспечивающих также управление данными, включая передачу и предоставление информации как внутри компании, так и вне ее. Некоторые существующие системы для непрерывного мониторинга и контроля промышленного оборудования позволяют управлять данными не только посредством самого программного обеспечения, но локальной сети, сети Интернет, но и также с помощью мобильных устройств — SMS-оповещения. Однако, подобный метод для персонала организации не всегда является удобным: предпочтение отдается уведомлениям через мессенджеры или оперативные чаты, с получением определенных сообщений с конкретными сведениями. Для уведомления об авариях промышленного оборудования разработан бот Telegram с использованием программной платформы Camunda. Приведенный в работе алгоритм системы уведомления об авариях может быть использован в различных, в том числе не производственных, сферах, для формирования уведомлений.

Ключевые слова: инженеринговые решения автоматизации, промышленное оборудование, системы аварийного оповещения, системы уведомлений, Camunda, бот Telegram.

Введение

Системы непрерывного мониторинга и контроля промышленного оборудования позволяют в реальном времени фиксировать состояние оборудования, идентифицировать причины простоя [1–2]. Рассматривается система сбора данных о работе обо-

рудования с устройством числового программного управления (УЧПУ), контроллеров станка, станочных терминалов. Мониторинг осуществляется посредством программного обеспечения, интернет-сервиса (web-диспетчер), коллективного пульта мониторинга, а также предусмотрен встроенный механизм СМС-оповещения персонала с использованием специального модуля, ос-

Станок	Событие	Начало	Окончание	Длительность
(ТЕСТ) МЕСМILL/ 411- 560/ ц.5/Расточ. станок	тест тэгов	17.02.2025 13:23:10	17.02.2025 13:23:11	00:00:01

Рис. 1. Модуль «Контроль событий»

нащенного SIM-картой. Модуль, в котором происходит сохранение записей об авариях промышленного оборудования, подключенного к системе непрерывного мониторинга, называется «Контроль событий». «Контроль событий» представляет собой таблицу, в которую записываются события, формируемые системой мониторинга промышленного оборудования (пример события, показан на рисунке 1).

По событиям в системе происходит рассылка оповещений для ответственных работников предприятия о простое оборудования и проблемах при производстве продукции. Рассылка оповещений происходит посредством отправки СМС-оповещений, пример содержимого СМС-оповещения представлено на рисунке 2.

На станке (ТЕСТ) Расточной
МЕСМILL (411-560) выставили
наладку

Рис. 2. Пример текста СМС-оповещения

Однако, механизм настройки рассылки подобных уведомлений ограничен и не обеспечивает необходимую гибкость, не позволяя выводить важную информацию, в особенности для механиков и ремонтников, например, протоколы подключения станка к системе мониторинга. Кроме того, для современных пользователей требования удобства больше отвечают уведомления через мессенджеры или оперативные чаты [3–4]. Также необходимо отметить финансовый момент — отправка СМС-оповещений происходит за счет GSM модуля и за каждую СМС взимается плата в соответствии с установленным тарифом от поставщика мобильных услуг. В связи с этим, возникает необходимость разработки инструмента с более гибкими настройками, для этой цели выбрана платформа Camunda.

Camunda — открытая платформа, представляющая собой набор библиотек на языке программирования Java, позволяющая создавать схемы бизнес-процессов, выполнять описанные процессы, создавать цифровой продукт по требованиям заказчика; используется для автоматизации бизнес-процессов [5–9]. В качестве источника данных для платформы Camunda используется база данных рассматриваемой текущей системы мониторинга и контроля промышленного оборудования предприятия (далее DataBase). Для взаимодействия DataBase с платформой предусмотрено использование дополни-

тельного плагина и модуля, позволяющих решать задачи по обслуживанию и администрированию.

Постановка задачи. Разработка системы уведомлений об авариях промышленного оборудования в виде Telegram-бота на базе программной платформы Camunda.

Методология

Особенности установки и настройки компонентов для работы Camunda. Для выполнения поставленной в работе задачи необходимо проделать подготовительные процедуры:

- 1) Установка Node.js и/или обновления до актуальной версии (в данной работе использована версия 18.20.3). Node.js — это среда выполнения JavaScript для создания серверных приложений, используются для запуска отдельных скриптов и приложений на JavaScript [10–13]. Node.js выступает в качестве базы для работы Camunda.
- 2) Установка и настройка плагина Camunda. Для установки плагина Camunda и прописывания его адреса в файлах конфигурации, копируется папка плагина по следующему пути: .../Industry/Server/Plugins, в конфигурационном файле appsetting.json необходимо раскомментировать свойства Plugins и поменять настройки:

```

"Plugins": {
  "CamundaPlugin": {
    "CamundaUrl": "http://localhost:9090",
    "SqlQueryOptions": {
      "ConnectionStrings": {
        "Default": "Provider=MsSqlServer;
        Server=localhost,1433;Database=
        DataBase;User ID=User;Password=User;"
      }
    },
    "HttpRequestOptions": {
      "Templates": {
        "MdcRest": {
          "Url": "http://localhost:9000/
IndustryRestManagement/",
          "Headers": {
            "Authorization": "Basic -----" //
указать пароль base64
          }
        }
      }
    }
  }
}

```

Создание и настройка бота Telegram. Для создания бота Telegram, через который будут отправляться сообщения, необходимо [14-18]:

1. Переход в диалог с инструментом для разработки чатов — <https://telegram.me/BotFather>.
2. Кнопка «Start» или ввод в диалоге команды /start.
3. Ввод команды /newbot, для создания нового бота.
4. Указание названия, которое будет отображаться в чате (ник бота) в списке контактов. Может быть любым, даже дублировать уже существующее.
5. Указание системного имени, которое будет ником после знака @. Обязательно должно быть уникальным, если имя уже занято, всплывет подсказка: «Sorry, this username is already taken. Please try something different».
6. После создания бота в чате получается токен. Токен бота Telegram — это индивидуальный цифро-буквенный код, который хранится в тайне для избежания получения доступа и использования злоумышленниками. Этот токен сохраняется, так как он используется для дальнейшей интеграции.

Написание и установка внешнего обработчика. Ниже приведен код (скрипт) внешнего обработчика на JavaScript:

```
import { Client, logger } from "camunda-external-task-client-js";
import { Variables } from "camunda-external-task-client-js";
import fetch from "node-fetch";
const config = { baseUrl: "http://localhost:9090/engine-rest",
use: logger }; // Конфигурация клиента Camunda
const client = new Client(config); // Создание экземпляра клиента
// Подписка на топик 'SendMessageTelegram'
client.subscribe("SendMessageTelegram", async function({
task, taskService })
{
// Получение переменных из процесса Camunda
const cDeptName = task.variables.get("deptname");
const cTextMessage = task.variables.get("textmessage");
// Асинхронная функция для отправки запроса и получения ответа
async function answerOk (url) {
```

```
let response = await fetch(url);
let data = await response.json();
return JSON.parse(JSON.stringify(data).replace(/
/g, " ")).ok;
}
// Отправка сообщения в Telegram и получение результата
let answer = await answerOk ('https://api.telegram.org/bot bot556...t0vR/sendMessage?chat_id=12345&text='+cDeptName+'%0A'+cTextMessage);
// Проверка результата и завершение задачи
if (answer == true) {
await taskService.complete(task); //complete task
}
else {
let answer1 = await answerOk ('https://api.telegram.org/bot556...t0vR/sendMessage?chat_id=67890&text='+cDeptName+'%0A'+cTextMessage)
await taskService.complete(task);
}
});
```

Этот код подписывается на задачи с топиком «SendMessageTelegram» в процессе Camunda и отправляет сообщения в чаты Telegram на основе полученных переменных из процесса. Топик — это название блока очереди в Camunda. В URL ссылке вводится токен бота Telegram. Этот код сохраняется в файле с именем SendMessageTelegram.js и помещается в папку, откуда обработчик будет работать.

Запуск скрипта производится в Node.js command prompt, последовательным вводом следующих команд:

- cd {путь к папке с внешним обработчиком}
- npm install -s camunda-external-task-client-js
- npm install node-fetch
- node .\TEST.js

В выбранной папке появляется несколько файлов с установленными модулями, библиотеками и конфигурационными файлами (рисунок 3).

Разработка диаграммы в Camunda Modeler. В качестве источника данных используется DataBase. Отбор

Имя	Дата изменения	Тип	Размер
 node_modules	07.05.2024 9:47	Папка с файлами	
 package.json	07.05.2024 9:47	Файл "JSON"	1 КБ
 package-lock.json	07.05.2024 9:47	Файл "JSON"	10 КБ
 README.txt	20.06.2024 10:13	Текстовый докум...	1 КБ
 SendMessageTelegram.js	09.07.2024 9:13	файл JavaScript	2 КБ

Рис. 3. Установленные модули

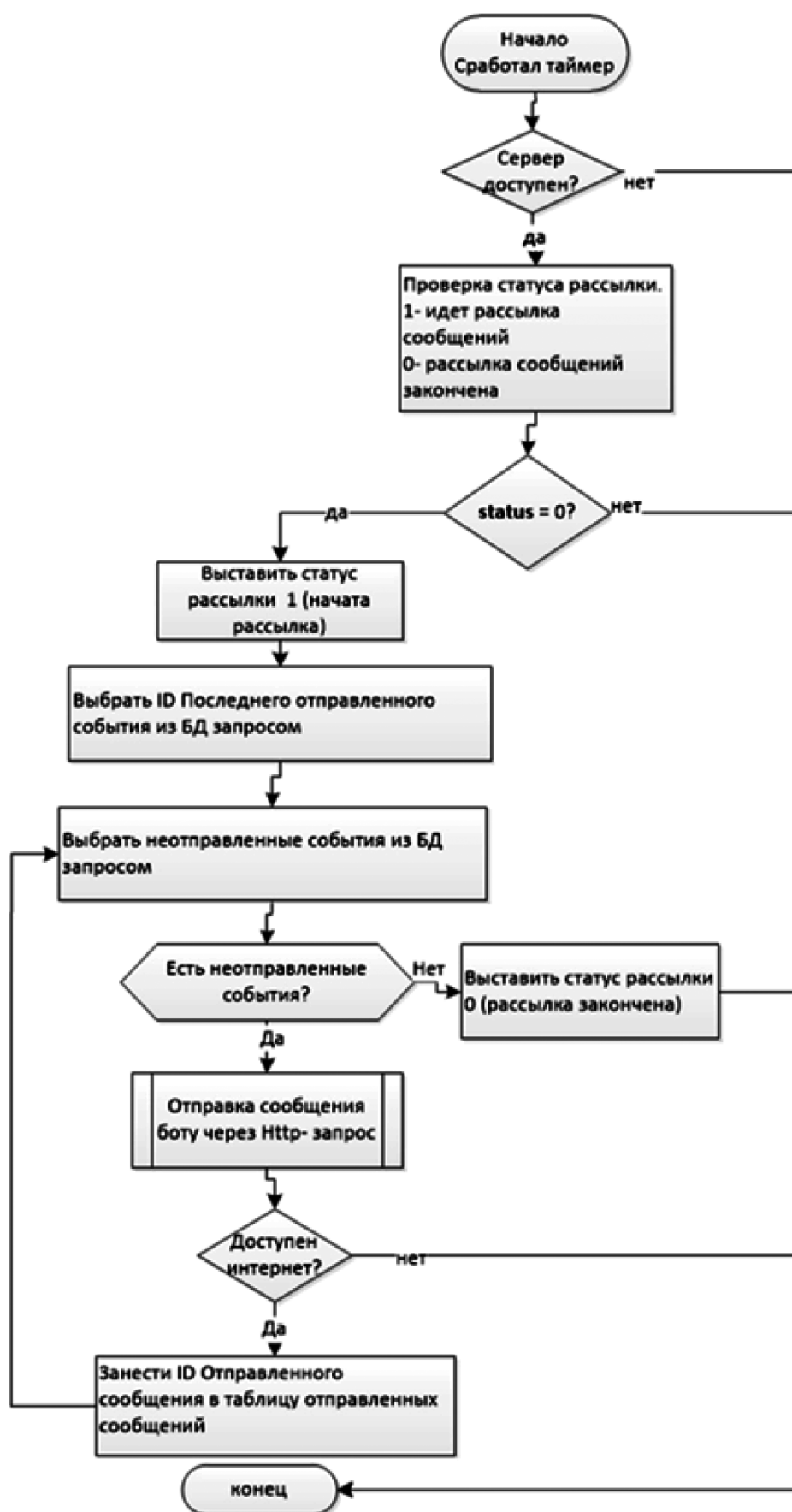


Рис. 4. Алгоритм уведомлений

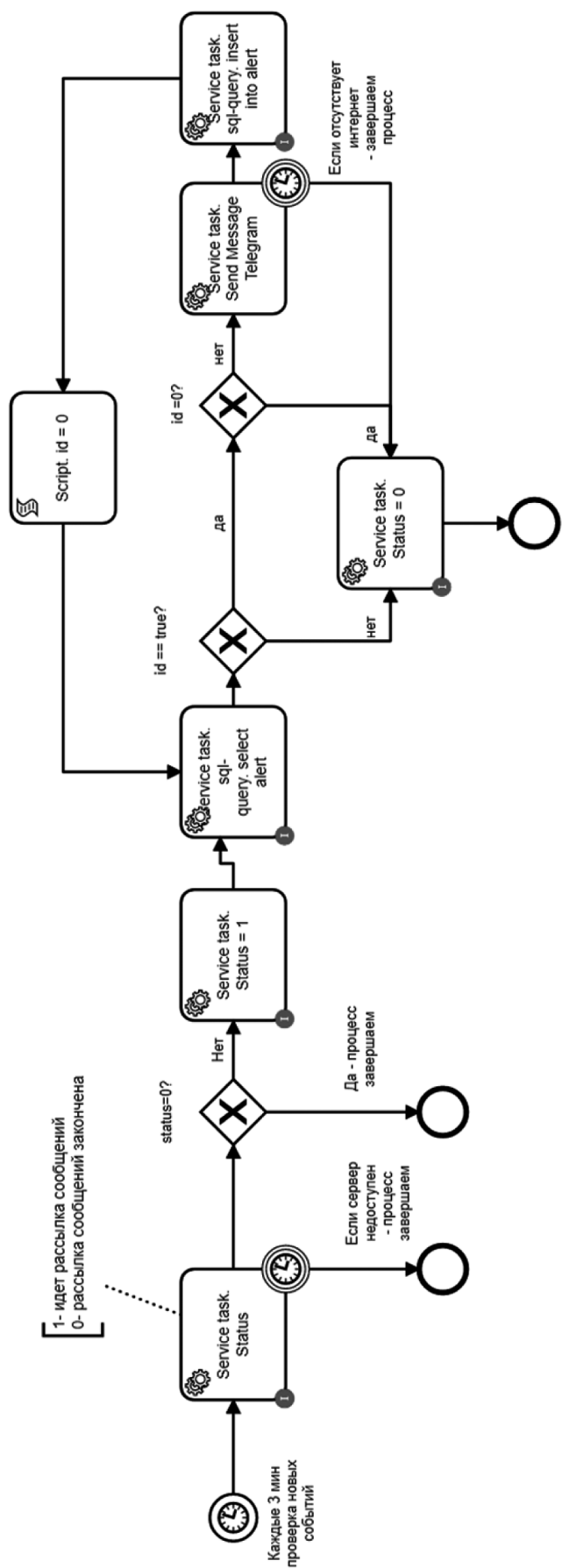


Рис. 5. Диаграмма системы уведомлений

необходимых уведомлений ведется по следующим условиям:

- Тип события «Авария».
- Дата возникновения — последние сутки (всё, что больше суток, считается неактуальным).

Для записи статуса рассылки и хранения последнего отправленного сообщения создаются отдельные таблицы в базе. Такое решение обусловлено поведением Camunda, после завершения процесса все переменные удаляются и хранить долгосрочные данные в них не получится.

Разработка системы осуществляется в Camunda Modeler. В построении диаграммы используются следующие компоненты:

- Start event и End event — определяют, где начинается и заканчивается процесс. В диаграмме требуется по крайней мере одно событие начала для создания экземпляра процесса.
- Gateway — используется для моделирования решения в процессе. Когда выполнение достигает этого шлюза, все исходящие потоки последовательности оцениваются в том порядке, в котором они были определены. Поток последовательности, условие которого оценивается как «истина», выбирается для продолжения процесса.
- Timer — блок, в котором задается время запуска события;
- Service task — используется для вызова служб. В блоки Service task в поле «SQL команда» прописываются sql-запросы к базе системы DataBase (АИС Диспетчера). Строку подключения к базе данных можно оставить пустой, т.к. имя шаблона автоматически подтянется из конфигурационного файла.
- Script task — это автоматизированная деятельность. Когда выполнение процесса достигает задачи скрипта, выполняется соответствующий сценарий.

Разработка алгоритма системы уведомлений. Алгоритм системы уведомления приведен на рисунке 4.

На основе приведенного алгоритма проектируется система уведомлений.

Результаты

На рисунке 5 показана спроектированная диаграмма системы уведомлений.

После проектирования диаграмма публикуется в Camunda Cocpit с помощью функции «Deploy diagram» для начала работы (рисунок 6).

Deploy diagram

Deployment name

Telegram-рассылка сообщений об аварии v

Tenant ID

Optional

REST endpoint

http://localhost:9090/engine-rest

Include additional files



Deploy

Рис. 6. Публикация диаграммы

После публикации необходимо перейти в Camunda Cocpit по адресу <http://localhost:9090/>, логин и пароль здесь по умолчанию Admin и Admin (рисунок 7).

CAMUNDA

Cockpit

Username

Password

Log in

Рис. 7. Страница авторизации

Далее открывается панель управления процессов (рисунок 8).

Данная панель позволяет видеть развернутые определения процессов. Необходимо выбрать нужный развернутый процесс.

Представление экземпляра процесса приводится на рисунке 9.

Это означает, что процесс уже начал работать.

Обсуждение

В качестве системы мониторинга и контроля промышленного оборудования рассматривалась система АИС «Диспетчер». Это универсальная российская система, разработчиком программного решения являет-

Camunda Cockpit

Processes

Decisions

Human Tasks

More ▾

Admin Admin

🏠

Dashboard » Processes

5 process definitions deployed

List

Previews

State	Incidents ▾	Running Instances ▾	Name ▾	Tenant ID ▾
✔	0	0	telegram рассылка сообщений об аварии v4	
✔	0	2	telegram рассылка сообщений об аварии v2	
✔	0	1	telegram рассылка сообщений об аварии	
✔	0	0	send_mes_teleg_admin	
✔	0	0	Process_1pbyy9x	

Рис. 8. Список процессов

Camunda Cockpit Processes Decisions Human Tasks More ▾					Admin Admin	
Dashboard » Processes » telegram рассылка сообщений об аварии v4 : Runtime						
Definition Version: 3 ▾ Version Tag: 2.0 Definition ID: telegram_alert_error_v4:3:6b3c0b6d... Definition Key: telegram_alert_error_v4 Definition Name: telegram рассылка сообщений об ав... History Time To Live: 1 day ✕ Tenant ID: null Deployment ID: 68640c3b-3a95-11ef-8735-00155d... Instances Running: • current version: 0 • all versions: 0	Activity Instance Statistics: on ● Process Instances Incidents Called Process Definitions Job Definitions Add criteria No process instances matched by current filter.					

Рис. 9. Список процессов

ся группа компаний «Цифра», позволяющая контролировать работу любого оборудования и рабочих мест, обеспечивающая объективными данными все системы управления предприятием [19]. Разработчиками АИС «Диспетчер» для взаимодействия с платформой Camunda были разработаны дополнительные плагин и модуль. Схема взаимодействия между пользовательским приложением АИС «Диспетчер» и платформой показана на рисунке 10 [20].

Принцип работы платформы Camunda заключается в том, что она запускается как отдельное приложение,

куда интегрируются процессорные приложения в виде war-файлов. Несмотря на то, что язык, на котором разрабатана платформа Camunda, это Java, для работы может быть использован любой язык программирования.

В данной работе была использована версия Node.js 18.20.3, в более ранних версиях Node.js может не сформироваться конфигурационный файл package.json. Если такое произошло, необходимо создать его самостоятельно и вписать в него следующий текст:

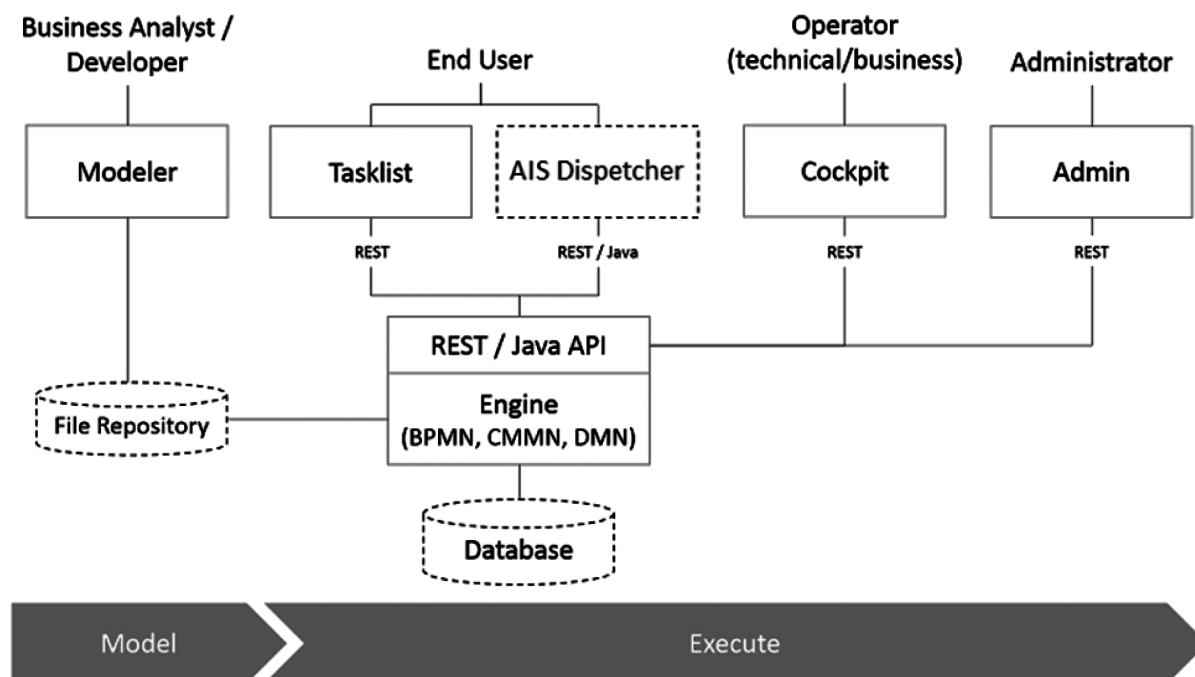


Рис. 10. Схема взаимодействия компонентов Camunda

```
{
  "dependencies": {
    "camunda-external-task-client-js": "^3.1.0",
    "node-fetch": "^3.3.2"
  },
  «type»: «module»
}
```

В итоге, в консоли отображается сообщение (рисунок 11).

```
Node.js command prompt - node .\SendMessageTelegram.js
C:\Users\aisdispatcher>cd c:\JS
c:\JS>node .\SendMessageTelegram.js
subscribed to topic SendMessageTelegram
```

Рис. 11. Подписка на топик

Данное сообщение в консоли о том, что внешний обработчик успешно подписался на топик Camunda.

Выводы

Был проведен анализ возможностей систем непрерывного мониторинга и контроля промышленного оборудования. Произведены необходимые начальные настройки платформы Camunda, получающей информацию из базы данных имеющейся на предприятии системы мониторинга оборудования, для реализации разработки системы уведомлений. Построен алгоритм работы системы уведомлений на базе платформы Camunda, осуществлена разработка системы уведомлений об авариях на предприятиях в виде Telegram-бота. Полученная эффективность работы системы уведомлений об авариях имеет важное значение для оперативного реагирования ремонтных служб.

ЛИТЕРАТУРА

- Кузнецов А.Е., Смирнова И.Г. Разработка автоматизированной информационной системы «Учёт заявок» / Экономика и социум, 2021. — №5(84), ч.2. — С.1202–1208
- Иванов Г., Селезнев Д., Каадзе А.Г. Автоматизация бизнеса: программы, инструменты и технологии / Статья на сайте газеты «Комсомольская правда», 2023 [Электронный ресурс] // URL: <https://www.kp.ru/guide/avtomatizatsiya-biznesa.html>
- Березин Е. Цифровая трансформация цементного завода (ч. 4): автоматические личные кабинеты и чат-боты / статья на сайте Хабр (бывший Хабрахабр) — русскоязычном веб-сайте в формате системы тематических коллективных блогов, 2021 [Электронный ресурс] // URL: <https://habr.com/ru/articles/553174/>
- Нигматуллин К.Р., Забродин А. В. Исследование и разработка чат-бота для оптимизации поддержки технологов системы «График исполнительного движения» в РЖД: выбор и обоснование платформы / Интеллектуальные технологии на транспорте, 2024. — № 1 (37). — С.61–72.
- Camunda — описание, примеры и инструкции на русском [Электронный ресурс] // URL: <https://camundarus.ru/>
- Автоматизация бизнес-процессов на open-source-технологиях (Camunda) / Информация на сайте компании ЛАНИТ — Би Пи Эм — Российской ИТ-компании, эксперта в области анализа, интеллектуальной автоматизации и трансформации корпоративных бизнес-процессов [Электронный ресурс] // URL: <https://lanitbpm.ru/camunda>

7. Camunda: тестируем модели процессов / статья компании SimbirSoft — компании, предоставляющей услуги по разработке и тестированию программных продуктов на сайте Хабр (бывший Хабрахабр) — русскоязычном веб-сайте в формате системы тематических коллективных блогов, 2022 [Электронный ресурс] // URL: <https://habr.com/ru/companies/simbirsoft/articles/657697/>
8. Dobby Guides Camunda. Часть 1/3 / статья на сайте telega.ph — инструменте для создания блогов и статей, 2023 [Электронный ресурс] // URL: <https://telega.ph/Camunda-CHast-13-09-05?ysclid=m3lrxi4lvx556476583>
9. Котов Д. Первый проект на Camunda: добавляем формы и автоматизацию / статья в блоге архитектора бизнес-процессов в Тинькофф [Электронный ресурс] // URL: <https://bpmn2.ru/blog/proekt-na-camunda-dobavlayem-formi-i-avtomatizaciю>
10. Трунин В. Путь программиста T-SQL / Самоучитель по языку Transact-SQL. Основы программирования для начинающих разработчиков. — Издательство: Info-Comp, 2020. — 204 с.
11. Официальная документация по работе с MSSQL / Информация на сайте Microsoft Learn — средстве для освоения продуктов и служб от разработчиков Microsoft, 2025 [Электронный ресурс] // URL: <https://learn.microsoft.com/ru-ru/sql/relational-databases/lesson-1-connecting-to-the-database-engine?view=sql-server-ver16>
12. Руководство по Node.js, часть 1: общие сведения и начало работы / перевод статьи Flavio Copes в блоге компании RUVDS.com, 2018 [Электронный ресурс] // URL: <https://habr.com/ru/companies/ruvds/articles/422893/>
13. Документация по модулю Node Fetch. Легкий модуль, который добавляет API Fetch в Node.js [Электронный ресурс] // URL: <https://www.npmjs.com/package/node-fetch>
14. Хватков Г. Боты в Telegram: какими они бывают, как работают и как сделать своего / статья на сайте Skillbox — российской образовательной платформы с онлайн-курсами, 2022 [Электронный ресурс] // URL: <https://skillbox.ru/media/marketing/boty-v-telegram-kakimi-oni-byvayut-kak-rabotayut-i-kak-sdelat-svoego/>
15. Как создать бота в Telegram: большая инструкция / статья на сайте Skillfactory — российской онлайн-школы IT-профессий, 2024 [Электронный ресурс] // URL: <https://blog.skillfactory.ru/kak-sozdat-bota-v-telegram/>
16. Telegram Bot API / Официальная документация Telegram API для управления ботом [Электронный ресурс] // URL: <https://core.telegram.org/bots/api#formatting-options>
17. From BotFather to 'Hello World' / Официальное руководство по созданию и программированию Telegram-бота [Электронный ресурс] // URL: <https://core.telegram.org/bots/tutorial>
18. Как создать чат-бота в Telegram / Информация на сайте Sber Developer — сайте технологий и отраслевых решений от Сбера для развития малого и среднего бизнеса, 2024 [Электронный ресурс] // URL: <https://developers.sber.ru/help/salutebot/telegram-integration>
19. Официальный сайт группы компаний «Цифра», разрабатывающей и внедряющей промышленные цифровые решения на базе собственной платформы, предоставляющей доступ к среде разработки платформенных приложений, развивающей индустрию роботизированного промышленного транспорта в горнодобывающей, нефтегазовой, химической и машиностроительных отраслях [Электронный ресурс] // URL: <https://www.zyfra.com/ru/about/>
20. Диспетчер — сайт многофункционального программного комплекса для оперативного контроля и повышения эффективности работы промышленных предприятий [Электронный ресурс] // URL: <https://intechology.ru/>

© Савельева Юлия Олеговна (Savelieva_yu_ol@mail.ru); Кулагина Надежда Алексеевна; Ваулин Владимир Иванович; Лукьянов Александр Александрович
Журнал «Современная наука: актуальные проблемы теории и практики»