DAIKIRI
Erklärbare Diagnostische KI für industrielle Daten

Project Number: 01IS19085 **Start Date of Project:** 01/01/2020 **Duration:** 30 months

Deliverable 6.1
Use Case Specification

Dissemination Level	Public
Due Date of Deliverable	Month 30, 30/06/2022
Actual Submission Date	Month 30, 24/06/2022
Work Package	WP6 — Use Cases
Task	T6.1
Type	Report
Approval Status	Final
Version	1.0
Number of Pages	10

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

This project has received funding from the Federal Ministry of Education and Research (BMBF) within the project DAIKIRI under the grant no 01IS19085B.

History

Version	Date	Reason	Revised by
0.0	01/06/2022	First draft created	Mohammad Sajjadi
0.1	03/06/2022	Draft revised	Bianca Preißler
0.2	08/06/2022	Draft revised	Jean Chorin, Leo Granger
0.3	10/06/2022	Draft revised	Martin Voigt
0.4	17/06/2022	Final version created	Mohammad Sajjadi
0.5	22/06/2022	Peer reviewed	Martin Voigt, Jonas Haupt
1.0	24/06/2022	Final version submitted	Mohammad Sajjadi

Author List

Organization	Name	Contact Information
elevait	Jean Chorin	jean.chorin@elevait.de
elevait	Leo Granger	leo.granger@elevait.de
elevait	Jonas Haupt	jonas.haupt@elevait.de
elevait	Bianca Preißler	bianca.preissler@elevait.de
elevait	Mohammad Sajjadi	mohammad.sajjadi@elevait.de
elevait	Martin Voigt	martin.voigt@elevait.de

Executive Summary

In this section, an overview of the use cases are given. Although, the initial cases are defined, they would be extended over the project runtime since 1) new research insights require more or other input, 2) research insights show that new results are feasible, and 3) new similar use cases come up which are helpful to validate the research results.

Contents

1 Introduction

1.1 Use Case 1: Smart Logistics

A timely delivery of any parts to production lanes is crucial to have an efficient assembling of products - for serial and project-based production. It is required to have the right parts in a sufficient amount (not less but also not too much) at the correct place at a given time. Therefore, a pro-active system is required which can predict the correct amount to deliver. However, there are a lot of obstacles to be solved, e.g., identification and elimination of anomalies, unclear production plans etc. Additionally, the results of the prediction need to be explainable and so judgeable if an AI-based system is doing well.

1.1.1 Target Group

There are 2 target groups as users of the system. First, the company and their logistic experts, which are responsible to deliver the parts to the production lanes of its customers in-time. Occasionally, these experts need to validate the usually accumulated amounts to deliver. Second, the controllers of the production company are reviewing the quantity of the delivered parts. Both experts need to have explainable predictions.

1.1.2 Operating Environment

Figure 1 gives an overview of the environment of the use case.

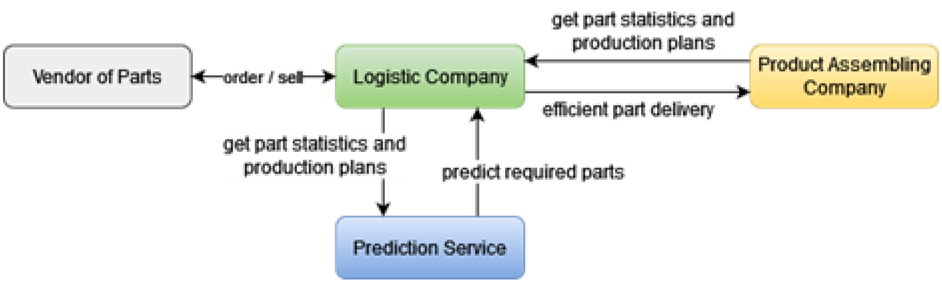

Figure 1: The use case environment for Smart Logistics

1.2 Use Case 2: elevait in:forms

Transmission of the forms filled out either manually or digitally is usually done via scanning or taking photos, which hinders simple automated reading of data. This repetitive typing of form data is also associated with a high susceptibility to errors.
elevait in:forms is an AI based software solution which is developed to support the customers by automating the extraction of relevant content from forms, in a reliable way.

1.2.1 Target Group

elevait in:forms targets every company that has to handle a lot of forms within it's business processes. It reduces the effort of checking and processing the forms and extracting relevant data out of them. This reduced effort pays out to the companies, as they are able to use the gained working time for other, more creative or relevant work.

1.2.2 Operating Environment

Figure 2 gives an overview of the environment of the use case.

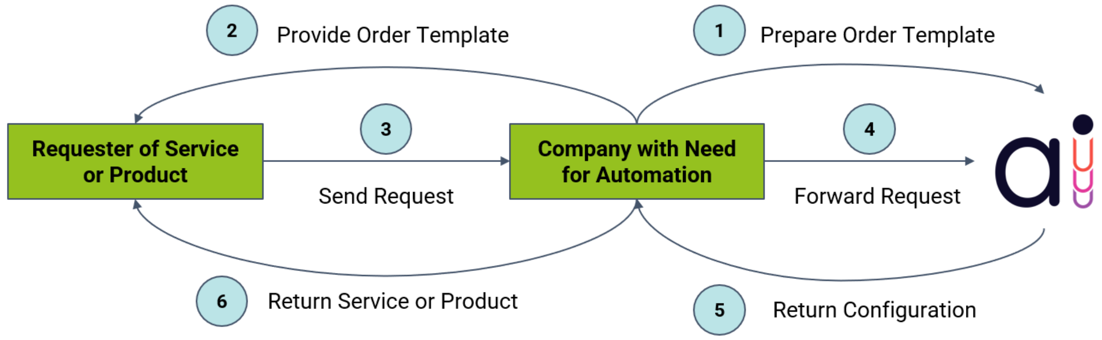

Figure 2: in:forms use case

2 Business Understanding

2.1 Use Case 1: Smart Logistics

In today's fast moving world, manufacturing and production are the true drivers of growth. The better, the leaner, and the smarter you make your factory, the more likely you'll succeed in the market. As a logistic company, you have to manage A-, B- and C-Parts of thousands of customers, based on smart logistics systems. In this use case, we focus on B- and C parts. In product assembly, B parts make up to 25% of the costs overall parts and have an amount of 10-40% of all parts assembled. C-parts have only about 5-10% of the value but make up to 70% of all parts. Typical C-parts are screws or nuts. Figure 3 shows this relation.

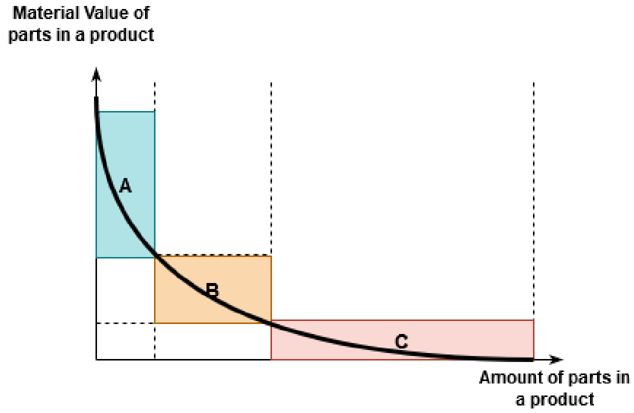

Figure 3: The relation between Material Value and Amount of parts, within a product

As a logistics inventory management company, you have the responsibility to provide the right parts:

- with the right quantity,
- with the right quality,
- at the right time,
- at the right place.

Unfortunately, it is not possible to get an useful forecast from customers on B- and C-Part level. If one is available, either timing of usage or quantities are not correct. With other words, the bill of material (BOM) of the customers are incorrect or incomplete in relation to B- and C-parts. The **main objective** is, to make use of the historic inventory data AND the future production plans of the customers in order to **predict** the required amount of B- and C-parts for the **next 2 replenishment cycles**. Since it is not feasible in one step and ILP is currently targeting classification tasks, the main business objective in the project is the **detection, classification and prediction of anomalies**

Evaluation strategy

- **Service Level Agreement** (SLA per assortment UUID):
  - Upper bound: 10% of reorder point value
  - Lower bound: 5% of reorder point value
- Evaluation of data points per assortment UUID
  - Good: data points inside the SLA
  - Bad: data points outside the SLA
- Evaluation of predictions per assortment UUID:
  - Good: assortment UUIDs where all predictions are inside the SLA
  - Bad: assortment UUIDs where all predictions are outside the SLA

2.2 Use Case 2: elevait in:forms

Even nowadays, a huge number of forms are still in circulation in a company or administrative context, e.g., to record orders, customer data, and surveys. The submission of the mostly manually and handwritten forms on a document base is mostly done as a scanned or as a photo, which hinders a simple automated readout. The necessity of manual typing of the corresponding information leads to high costs and a high resource usage. As a result, the number of forms to be processed cannot be scaled arbitrarily, especially in the times of increasing business development and the need for more customer service quality. The number on customer interaction is steadily increasing, while the number of people doing the work is decreasing due to the high monotony of this type of activities, which ends in decreasing motivation and frustration of employees. The repetitive typing of form data is also prone to errors.
elevait in:forms provides extraction and validation of information from forms, web pages or format-based layouts. in:forms is also capable of:

- Recognition and classification of documents such as scans, faxes or PDFs
- Processing of machine and handwriting
- Check for presence of signatures or comments
- Automatic application of freely definable validation or plausibility rules to business documents

Besides extracting and validating information out of a form, elevait offers the opportunity to actually capture company knowledge with a rule editor to enrich the extracted information. Usually, the extracted information is marketing-dictated to make it understandable to end customers. Whereas the extracting employees enrich the information in a format, that the endsystem, e.g. the production system, can actually make sense out of it. elevait enables the employee to define this data transformation steps into a standard to actually automatize holistically.

2.2.1 Problem

The output quality of the model fluctuates for different customers, different structures of processed forms, as well as data extraction models. A highly anticipated factor in this matter is the user behaviour, which significantly varies across the customers. For now, there are no precise information and metrics on AI-related user misbehaviour, which decreases model performances and increases the need in problem-specific model configurations. This leads to more configuration costs, longer onboarding periods and less reusable software progress. (Figure 4)

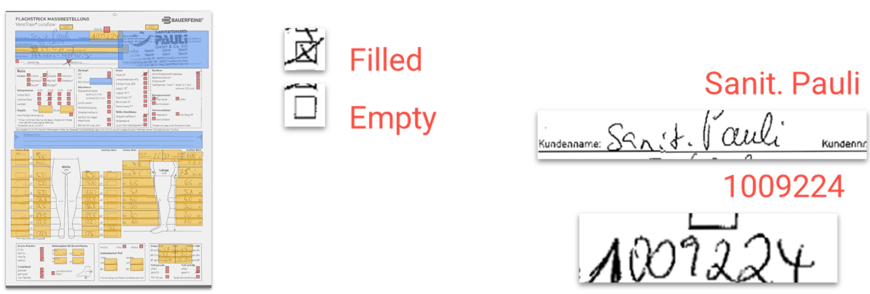

Figure 4: Examples of handwritten fields, and the extracted data

2.2.2 Objective

Finding the reasons for fluctuations and changes, by examining the extracted data and answering questions such as:

- Is there any meaningful relation between requesters and validation errors?
- Are there common mistakes occurring on the forms?

This would allow to determine common and reoccuring errors on the user side, communicate them to the customer and promote AI-friendly user behavior. Edge case solving has to involve more and more the interaction with users and not only on the design of AI system, which carries high risk in creating specific overfitted solutions for individual customers.

3 Data Understanding

3.1 Use Case 1: Smart Logistics

Overview of the resource hierarchy

- Hierarchy of resources (Figure 5):
  - each customer has multiple production lines **(sites)**
  - each site consists of multiple boxes **(locations)**
  - each location contains multiple fastening elements of a specific type **(assortment UUID)**
  - each assortment UUID can be distributed over multiple boxes **(box number)**

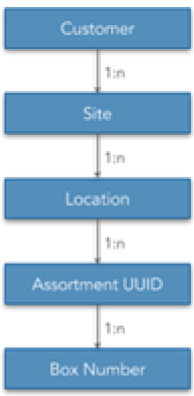

Figure 5: Hierarchy of resources

- **Determing the content of a certain box**
  - Known weight of a single assortment element (assortment UUID scale weight)
  - Amount of elements inside a box calculated by total weight
- **2 customers:**

  - data with different files and structures of about 3 years
  - each of about 130k entries

- **Identify and explain 3 anomalies (Figure 6):**
  - The replenishment is higher / lower than the quantity delivered
  - The replenishment was carried out without prior delivery
  - Minimal stock changes

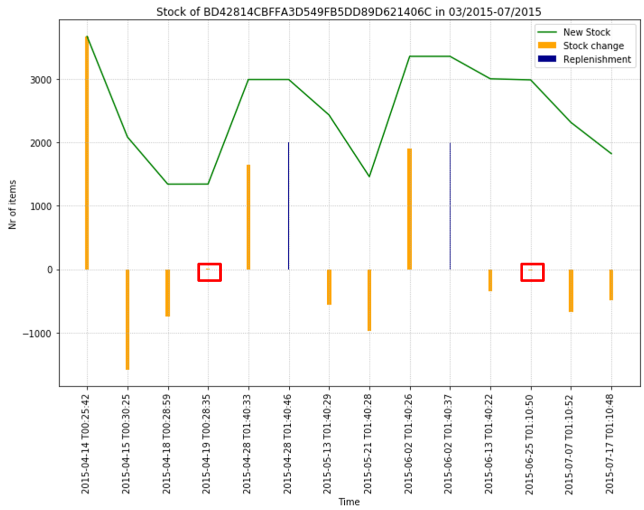

Figure 6: Example anomaly for the class **minimal stock change**

3.2 Use Case 2: elevait in:forms

The data of in:forms in stored using W3C Semantic Web standards, mainly RDF. Its data schema, depicted in (Figure 7) is described using RDFS and SHACL from the same standard family. The blue frame contains classes and their relation to describe an incoming form document using the class `edm:Document` and its `edm:DocumentPages`. Each page comprises some `edm:BoundingBoxAnnotation` which are rectangles where the content was extracted from. These are taken over from the template `edm:Document` shown in the green frame. These annotations are done be the domain expert during setup time in order specify the semantics to be extracted. In addition, the users can define business rules using `sh:NodeShape` from SHACL. These are executed on the extraction results generically defined as `gdm:Resource` and `gdm:GdmResourceAttribute`. If the business rules change the data model, each change is audited using `gdm:Order` and some relations depicted in the orange frame.

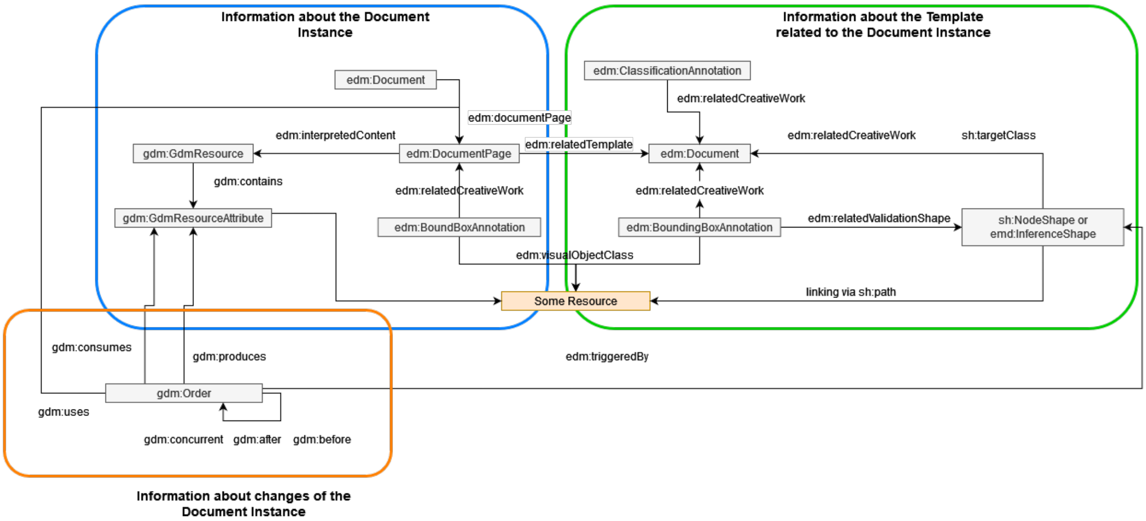

Figure 7: Overview of the Knowledge Graph of processed form

The data is not usable directly for the DAIKIRI pipeline since a) we can not use personal data and b) we want to show the automated identification of proper semantic classes. Therefore, we took the document data of one customer of one full month (about **62k** documents) and created a dedicated tabular data set as CSV file. In this, we just took over the confidence values of each extracted value (range **0** to **1**, the higher the better) and if a validation rule is not assigned (**NULL**), not hit (**0**) or hit (**1**). It lead to more than 3000 columns of the CSV file.