
Socially-acceptable
Extended Reality
Models and Systems

# D5.1 SERMAS Data Collection and Analysis
# Date 31/03/2023

Funded by
the European Union

| DELIVERABLE INFORMATION | |
|---|---|
| **Deliverable leader** | TUDa |
| **Document type** | DATA |
| **Document code** | D5.1 |
| **Document name** | Data collection and analysis |
| **Work Package / Task** | WP5 / T5.1 Data collection and analysis |
| **Delivery Date (DoA)** | 31/03/2023 |
| **Actual Delivery Date** | 31/03/2023 |
| **Reviewers** | L. Capra (SPXL)<br>L. Sabattini (UNIMORE)<br>L. Vigano (KCL)<br>V. Villani (UNIMORE) |

| DELIVERABLE HISTORY | | | |
|---|---|---|---|
| **Date** | **Version** | **Author** | **Summary of main changes** |
| 26.03.2023 | 0.1 | D. Petrak (TUDa)<br>T. Tran (TUDa) | Drafting the ToC and contribution distribution |
| 12.03.2023 | 0.2 | T. Tran (TUDa)<br>L. Capra (SPXL)<br>F. Zurolo (POSTE) | Adding content and details for the sections; minor changes in the outline; collecting contributions from other partners. |
| 24.03.2023 | 0.3 | T. Tran (TUDa)<br>D. Petrak (TUDa) | Collecting contributions and reviews from other partners; writing |
| 27.03.2023 | 0.4 | T. Tran (TUDa) | Finalizing the draft of the deliverable, without data statistics |
| 29.03.2023 | 1.0 | T. Tran (TUDa) | Finalizing the draft of the deliverable, updating with data statistics |

| DISSEMINATION LEVEL | | |
|---|---|---|
| **PU** | Public | **x** |

**SERMAS partners**

**Disclaimer**

This project has received funding from the Horizon Europe programme under the Grant Agreement 101070351.
Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.

# Public Executive Summary

This deliverable

- Presents the overall data collection process from designation of the data collection process and platform to the collected data statistics and analysis.
- Focuses on analysing the requirements from the project goals and the SERMAS pilots motivating our data collection.
- Describes the data collection process from participant recruitment, tasks of participants, annotation process and the progress so far.
- Describes the data collection platform designed by TUDa and developed by SPXL.
- Reports the statistics of the collected data by the submission time.
- Summarises our findings and lesson learned.
- Plans on the follow-up work.

# Table of contents

# List of Figures



# List of Tables

# 1. Introduction

This deliverable focuses on the conversational capabilities of the SERMAS XR agent. In particular, it presents the overall data collection process from designation of the data collection process and platform to the collected data statistics and analysis. The data collection platform will be publicly released in a scientific publication for making dialogue data collection more feasible in future research. The collected dialogue data will be used to facilitate training and testing the verbal communication interaction of the SERMAS agent. We also consider multimodality in the collected dialogues with the introduction of non-verbal communication signals attached to verbal communication in the form of textual description in this deliverable.

In the following sections, we first present the acronyms and terminology used in this report (Section 2). Next, in Section 3, we extensively analyse the requirements of SERMAS and the use cases with regards to verbal communication. We relate the requirements to existing publicly available dialogue datasets. From which, we present motivation for collecting a new multimodal dialogue data annotated with user feedback. In Section 4, we start with a detailed description of our data annotation process including our participants, dialogue collection strategies, the recruitment process and finally the list of dialogue tasks to be collected. The data collection platform will be fully described in the Section 5. Section 6 provides the metadata and statistics of the collected data. Afterwards, we present our lesson learned in this deliverable (Section 7). Lastly, we conclude the deliverable with the achievements and follow-up work (Section 8).

It is worth noting that we are working on publishing the data as a conference paper and possibly in further scientific papers where we enrich the textual description of non-verbal communication with other modalities such as sensing signals (WP4), images and/or speech.

# 2. Terminology

In order to facilitate the reader's understanding, we present the terms and their definitions used in this report (Table 1).

Table 1- Terms and definitions

| Term | Acronym | Definition |
|---|---|---|
| Data collection platform | DCP | the web-based platform developed by SERMAS for collecting dialogue data, which supports synchronous messaging between users |
| Application programming interface | API | the communication interface between components in the data collection platform |
| Task-oriented dialogue | TOD | focuses on supporting users to complete a particular goal |
| Document-grounded dialogue | DocDial | set for providing users information in documents via natural language interactions |
| Open-domain dialogue systems | OpenDial | try to engage users in open topics |
| Annotator | | a human (in the 1st collection version) -who plays the role of an agent or a user (of the agent) |
| User | | the user who is going to interact with the agent |
| Agent | | the SERMAS agent who is going to support User in completing tasks, providing information, and etc. |
| Intent | | the goal of an input from the user, such as getting access to a building, seeking information of a product/service |
| Slot | | the attribute types or properties that are required to fulfil user's intent, such as name of the user for building access or name of the internal inviter |
| Slot value | | the actual attribute value of a slot, such as "Paul" as the name of the user/speaker |
| Turn | | a dialogue interactive unit, contributed by one speaker in the dialogue |

# 3. Requirement Analysis

This section gives an overview of how we derive the requirements of the module in charge of dialogue management (see D3.1, module 14. *Dialogue management*) from the project goals and the SERMAS pilots, leading to our decision in which data to be collected and how to collect them.

## 3.1. Requirements based on Project Goals

In the scope of verbal communication, SERMAS aims at providing general tools for building ue agents that can be adapted to different domains. To realize this goal, we cover **three main paradigms of dialogues** in our data collection. These paradigms include:

- Task-oriented dialogue systems (TOD), which focus on supporting users to complete a particular goal
- Document-grounded dialogue systems (DocDial), which set for providing users information in documents via natural language interactions.
- Open-domain dialogue systems (OpenDial), which try to engage users in open topics.

While most related work studies these paradigms independently, these dialogue modes can potentially be intertwined together in the same dialogue session, as easily happened in human dialogues.

Since SERMAS's XR Agents are not limited to verbal communication but also consider non-verbal interactions[1], we also include the collection of ***potential non-verbal communication signals*** in our dialogue data. We consider ***emotions and user gestures*** while interacting with the agent as additional signals that can help the system to generate more engaging and socially acceptable responses.
Besides collecting the possible inputs that can be forwarded to our dialogue agents and the verbal responses to users, we also collect **potential physical interactions** that can be done by the SERMAS agents in the form of textual descriptions, inspired by Roitberg et al., (2014).

While including non-verbal communications can improve social acceptance and user experience to some extent, the abilities to ***correct mistakes*** and to ***adapt to new information*** of an agent should be considered as well, particularly in the real-world deployment. We, thus, collect the feedback interactions from users that allow continual learning to improve from such feedback. We note that the feedback collection is planned in this deliverable, but the data will be fully collected in the future due to the time constraints and the connection to dialogue models to be delivered in D5.2.

## 3.2. Requirements based SERMAS Pilots

In this section, we analyze the SERMAS pilots from our industrial partners that are described in Deliverable 2.1.

[1] For more details regarding non-verbal communication, we refer the readers to D4.1.

### 3.2.1. DW Security Agent

The first pilot by DW, which is a security agent, intentionally supports human instructors in security training for people working in media. We do not consider collecting data for the DW agent because the pilot scope falls into a different type of interaction, assessing answers by the trainees given multiple choice questions. For this pilot, we aim at generating alternative answers for a given question and the correct answer to the question. The generated answers are supposed to be under the scope of the question, really related to the correct one and sometimes hard to differentiate from the correct one without a deep understanding of the question's content. We will discuss this in later deliverables.

### 3.2.2. POSTE Post Office Agent (POA)

The Post Office Agent (POA) pilot from POSTE consists of 6 different tasks related to postal services. We collect the task descriptions, the required input for task completion as well as the expected output fields (Table 2).

For each task that is expected to be performed by the agent, we require the following information:

- Task id: task indexing
- Task: name of the task
- Related requirements: requirements related to the selected task, from D2.1
- Description of the service: task description
- Slots / information required to complete the task: slots or information that are required for the agent to complete the task; whose values should be provided by a user
- Information to be provided by a user: the description of each required slot/information
- Internal data: the data that can be retrieved from an internal database; information from the business that the agent should have access to
- Output data: information that the agent is expected

Table 2 – Task descriptions and data requirements for POSTE Post Office Agent (POA)

| **Task id** | | **POA01** |
|---|---|---|
| Task | | Parcel choice |
| Related requirements | | · FR_POA08: Detect object<br>· FR_POA16: Information procedure |
| Description of the service | | The user enters the Post Office to send a parcel and requests assistance from the POA for the choice of the postal product according to the weight, the shipping time and the destination |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| x | type of parcel | Type of parcel: destination, weight, packaged/without, shipping times |
| x | destination | Italy, foreign countries -> location |

| | | |
|---|---|---|
| x | weight | Lightweight up to 5kg<br>Average between 5-20kg<br>Heavy from 20-30 kg |
| x | packaged/without | · yes/no |
| x | shipping time | · delivery time within 4-6 days<br>· times 1-3 days |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | Parcel Product Description | Check among all the products which is the one that corresponds to the characteristics indicated by the user |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | parcel product name | Name of the specific postal product |
| x | parcel Product Description | Package characteristics |
| x | shipping procedure | Procedure for shipping the parcel |

| **Task id** | | **POA02** |
|---|---|---|
| Task | | Request ticket |
| Related requirements | | · FR_POA17: Release ticket number |
| Description of the service | | The user specifies the service and the corresponding ticket is issued |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| x | description of service | Type of service: shipping a parcel, pay a bill etc<br>e.g., parcel shipping, bill payment |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | ticket number | The POA interfaces with the ticket issuing system |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | ticket number | Ticket number associated with the service |

| **Task id** | **POA03** |
|---|---|
| Task | Commercial proposal |
| Related requirements | · FR_POA11: Promotional customized messages (cluster users)<br>· FR_POA12: Promotional messages<br>· FR_POA13: Promotional messages (recognized user) |
| Description of the service | The POA shows commercial proposal |

| **Slots / information required to complete the task** | | **Information to be provided by a user** |
|---|---|---|
| Required | Data type | Data description |
| x | user face | Face of the user standing in front of the camera |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | age | The POA calculate the age and interfaces with the Content system to release the commercial proposal |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | commercial proposal | Content fo the specific commercial proposal based on the age |

| **Task id** | | **POA04** |
|---|---|---|
| Task | | phone recharge |
| Related requirements | | · FR_POA14: Payment services |
| Description of the service | | The user wants to recharge a phone |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| x | number, import, phone provider, | Data needed to complete operation |
| x | phone number | + 39 xxx xxxx (Italian phone ?) |
| x | import | 10 €, 20 €, 30€ |
| x | phone provider | Poste mobile, TIM Vodafone, WindTre, Fastweb<br>TIM<br>Vodafone<br>Wind<br>3<br>three<br>CoopVoce<br>Carrefour UNO Mobile<br>Poste Mobile<br>Daily Telecom<br>A-mobile<br>Conad INSIM<br>MTV Mobile<br>Fastweb Mobile<br>Telepass Mobile<br>BT Mobile<br>Digitel Italia<br>Digi.Mobil Italia<br>Smart Pinoy<br>Tiscali Mobile<br>Noverca<br>ERG Mobile |

| | | |
|---|---|---|
| x | input card | The user insert the card in the card reader. The payment takes place through the POS, to understand how to simulate the payment |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | number, import, phone provider, | The POA interfaces with payment system and pass this information |
| x | outcome operation | if the data are corrects the POA ask to insert the card in the POS |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | operation outcome | The user visualizes the outcome of the operation |

| **Task id** | | **POA05** |
|---|---|---|
| Task | | Q&A |
| Related requirements | | · FR_POA15: Question answering |
| Description of the service | | The POA provide information about a generic financial product:<br>Postepay evolution<br>Linea Protezione Patrimonio |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| x | question | The user asks a specific information about a product |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | document | POA extracts the information from a specific document |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | response | The user visualizes the response extracted from the document |

| **Task id** | | **POA06** |
|---|---|---|
| Task | | Bill payment information |
| Related requirements | | · FR_POA16: Information procedure |
| Description of the service | | The user enters the Post Office to pay a bill and requests information from the POA |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |

| | | |
|---|---|---|
| x | Type of bills | Type of bills: Pre-printed form (code type 896/674) |
| | Other bill types | PagoPA, Empty form or MAV |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | Bill Form Description | Check among all types of bills forms is the one that corresponds to the characteristics indicated by the user |
| x | Information in the bill | The infos are: payer data, creditor data, amount and the pre-printed code(896). |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | Bill Form name | Name of the specific postal bill form |
| x | Bill Form Description | Bill form characteristics |
| x | Payment procedure | information for the filling form procedure and ticket |

### 3.2.3. POSTE Receptionist Agent

The Receptionist Agent (RA) pilot from POSTE consists of 7 different tasks related to reception (Table 3).

Table 3 - Task descriptions and data requirements for POSTE Reception Agent (RA)

| **Task id** | | **RA01** |
|---|---|---|
| Task | | Access the building to participate to meet a host |
| Related requirements | | · FR_RA04: Greetings to the visitor<br>· FR_RA06: Provide brief help<br>· FR_RA07: Request understanding |
| Description of the service | | The visitor (who is not a POSTE employee) has to participate in a meeting in a POSTE building. This interaction will happen the day of the meeting some time before the meeting starts. All the information about the meeting room, start/end time, guest name, etc. are available through the QR code that the guest will receive in the invitation for the meeting beforehand the meeting. |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| x | QR code | Contains: Host name, Host email, Meeting room identifier, Guest name, Meeting date and time |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | Host name | Name of the employee |

| | | |
|---|---|---|
| x | Alternative host name | Optional alternative host name |
| x | Host email | The host email corresponds to the account on the Microsoft Teams server in order to accommodate the guest confirmation call |
| x | Alternative host email | The alternative host email corresponds to the account on the Microsoft Teams server in order to accommodate the guest confirmation call |
| x | Meeting data and time | Date and time of the meeting |
| x | Meeting room identifier | Unique identifier of the meeting room |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | Verification call | The system will set a videocall to let the host to visually inspect the guest and authorize the access |
| x | Confirmation to open the turnstile | This is a signal toward the internal system that controls the turnstile to let the guest to enter |

| **Task id** | | **RA02** |
|---|---|---|
| Task | | Verify guest identity |
| Related requirements | | · FR_RA12: User verification |
| Description of the service | | In this task a two-factor authentication: one factor is the QR code the guest will exibith the second factor is the verification that the guest has the control of the email (from the details of the meeting) the ooleanon has been sent to |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| x | QR code | Contains: Host name, Host email, Meeting room identifier, Guest name, Meeting date and time |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | Click on the authenticate link in the email | Once the guest has shown the QR code we want to verify that he/she is really who he/she is expected for the meeting. For this purpose, an email is sent to the guest address containing a link that must be followed within a few minutes time span. |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | Whether or not guest has been recognized | These variable states whether the guest has been recognized |

| **Task id** | | **RA03** |
|---|---|---|
| Task | | Handle the video call between the agent and the host(s) to authorize the access to the building |
| Related requirements | | · FR-RA11: Video-call |
| Description of the service | | This task will be started by the agent and will handle the short dialogue between it and the guest. The agent will provide the information about the meeting, the guests who have already been recognized and will ask for the authorization to open the turnstile. The guest can: authorize the access, refuse the access or delay it too late. |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| x | Answer from the host(s) (yes, no, delay) | Answer from the host(s) to whether or not authorize the access or delay it |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | Host name | Name of the employee |
| x | Verified information coming from the QR code | Contains: Host name, Host email, Meeting room identifier, Guest name, Meeting date and time |
| x | Alternative host name | Optional alternative host name |
| x | Host email | The host email corresponds to the account on the Microsoft Teams server to accommodate the guest confirmation call |
| x | Alternative host email | The alternative host email corresponds to the account on the Microsoft Teams server to accommodate the guest confirmation call |
| x | Meeting data and time | Date and time of the meeting |
| x | Meeting room identifier | Unique identifier of the meeting room |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | Enable the QR code scanning on the turnstile for the given time | The QR code scanner will be enabled on the time it is authorized for, that is, in the prefixed meeting time or for the time for which it has been rescheduled |

| **Task id** | **RA04** |
|---|---|
| Task | Provide additional safety information |
| Related requirements | · FR_RA17: Mandatory safety brief<br>· FR_RA18: On request safety info |
| Description of the service | Under circumstances (e.g., COVID pandemic, area of the building closed for the access etc.…) before allowing the turnstile to open |

| | | |
|---|---|---|
| | | the system must provide additional safety information to each guest |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | Safety information video or audio | Safety information video or audio to be provided to guests |
| x | Guest has been recognized and the host has authorized the access | To start this task the guest has been recognized and the host has authorized the access |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | Safety information video or audio | Safety information video or audio to be provided to guests |

| **Task id** | | **RA05** |
|---|---|---|
| Task | | Greetings to the visitor |
| Related requirements | | · FR_RA04: Greetings to the visitor |
| Description of the service | | The RA greets the visitor using the most appropriate expression for the identified user class |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| Required | Approaching user | The user approaches the agent or otherwise manifests the intention to start an interaction |
| Required | User age range | Age range estimated by the user's shot |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| | N/A | |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | Greetings/welcome | The RA greets the visitor using the most appropriate expression for the identified user class |

| **Task id** | **RA06** |
|---|---|
| Task | Provide brief help on user's indecision detection |
| Related requirements | · FR_RA05: Detect user's indecision |

| | | |
|---|---|---|
| Description of the service | | The RA should be designed to provide helpful hints and tips to address common issues or to guide the visitor if they get stuck. |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| Required | User's indecision | A person interacting with the RA looks confused and in need of assistance |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| | N/A | |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |
| x | Brief help | A list of the services that RA can give help on (to enter the building for a meeting, provide directions for a given meeting room/office, provide safety information) and how to make a request |

| | | |
|---|---|---|
| **Task id** | | **RA07** |
| Task | | Guiding |
| Related requirements | | · FR_RA14: Instructions for reaching a location<br>· FR_RA19: Guiding |
| Description of the service | | The RA will provide a "carry-on" module to guide the visitor through the buildings until they reach the meeting place.<br>The module will be able to retrieve the target destination, orientate itself, follow the best path towards the destination, and alert the visitor if he/she has taken the wrong path; it will also provide additional information on the way to the destination. To ensure safety for both humans and it, the module will avoid any obstacles, both expected and unexpected. |
| **Slots / information required to complete the task** | | **Information to be provided by a user** |
| Required | Data type | Data description |
| x | Target destination | Identifier of the room to reach |
| **Internal data** | | **Information from the business that the agent should have access to** |
| Required | Data type | Data description |
| x | Building map | A map of the building |
| x | Current position | The position the agent is in right now |
| | POI | Additional information along the path |
| | User's position | Position of the user |
| **Output Data** | | **Information that the agent is expected to convey in its response** |
| Required | Data type | Data description |

| | | |
|---|---|---|
| x | Direction | Shows the direction to take based on the current position and the destination to reach |
| | Additional infos | Shows additional infos along the path, if any |

We note that the Section 3.2.2 and 3.2.3 are task requirements from POSTE, which may not relate to verbal communication such as RA03 – handling the video between the RA and the host, which can only be done by an internal video conference system used in POSTE. We thus will remove and merge some tasks to define intents (goals) that will be considered in the verbal communication module and in our data collection. More details about the final tasks or intents to be completed by the agent will be discussed in Section 3.4.

## 3.3. Analysis of Existing Datasets and Motivation

Existing datasets unfortunately do not fully support our pilots, because there are no publicly available datasets that cover dialogues in postal services. Furthermore, the available TOD datasets are mostly textual dialogues without considering non-verbal communication signals. While existing emotion-annotated multimodal dialogue datasets target open-domain dialogues, our data covers multiple paradigms of dialogues and from the analysis of production pilots as well as includes non-verbal communication signals.

## 3.4. Dataset Collection – Annotation Details

To not restrict potential research topics, our desired dataset not only include labels for intents and slots (which are necessary to train task-oriented dialogue systems), but also for errors in dialogues/responses, user feedback (error types and/or free-form feedback), and non-verbal communication signals (such as emotions, body gestures or physical interactions). Non-verbal signals will be collected as predefined textual labels in the current phase. We will enrich the dataset with available open-source emotion/gesture/physical interaction datasets in a post-processing step because the lists of possible non-verbal signals are to be finalized by WP4 during the SERMAS project.

To the best of our knowledge, such a dataset (feedback-annotated task-oriented dataset with non-verbal signals) is not publicly available yet. Therefore, this dataset will be a valuable contribution to future research on dialogue systems and extended reality dialogue systems.

In Table 4 (below), we present the data schema, list of intents and slots, that are collected in this deliverable in the YAML format. The list of intents is derived from the pilot tasks. For each intent, we define:

- Intent: name of the intent
- Description: description of the intent
- Original_task: mapping to the respective pilot task
- Required_slots: a list of requested slots for the task completion. For each slot, we collect the following information
    - Name: name of the slot
    - Description: description of the slot
    - Is_categorical: whether a slot will be restricted in a set of categories
    - Possible_values: a set of values that can be selected for the slot

- Optional_slots: a list of optional slots for task completion, a task can also be completed without these slots
- Result_slots: a list of slots or information needed to be considered by the agent when generating a response to a user
- Optional_result_slots: similar to optional_slots but are considered when generating a response

We note that the greetings at the beginning and the end of a dialogue can be considered as OpenDial (intent – greeting). For the intent "greeting", we include a slot named "user_class" to indicate the age generation of the user, which can used for user-aware response generation. In this data, we consider four major generations, including *boomer*, *gen_x*, *millennials*, and *gen_z*. Although we only consider greeting as OpenDial in this data collection, we can generally augment our data with existing open-domain datasets or with dialogues generated by pre-trained dialogue models.
The main parts of dialogues are usually DocDial (intent – question answering) or TOD (the rest). For DocDial, a particular document is considered in the dialogue as a reference for providing information to the user.

Table 4 - List of intents and slots in our data

YAML

```yaml
- intent: greeting
description: A dialogue needs to be started by a proper greeting. To do so,
the system needs information about the user class
original_task: RA05, RA06
required_slots:
- name: user_class
description: the user class
is_categorical: yes
possible_values:
- boomer
- gen_x
- millenials
- gen_z
optional_slots: {}
result_slots: {}
optional_result_slots: {}
- intent: request_parcel_choice
description: The user enters the Post Office to send a parcel and requests
assistance from the POA for the choice of the postal product
according to the weight, the shipping time and the destination
original_task: POA01
required_slots:
- name: destination
description: country of destination, national or international
is_categorical: yes
possible_values: []
- name: weight
description: weight of the parcel
is_categorical: yes
possible_values:
- light_weight
```

```
- average_weight
- heavy_weight
- name: required_package
description: yes/no package
is_categorical: yes
possible_values:
- true
- false
optional_slots:
- name: delivery_option
description: Express or standard delivery
is_categorial: yes
possible_values:
- express
- standard
result_slots:
- name: country_of_destination
description: check among all the products which is the one that corresponds to
the characteristics indicated by the user
- name: parcel_product_name
- name: parcel_product_description
- name: parcel_product_shipping_procedure
- name: shipping_time
description: depending on delivery option and whether the destionation is
national or in another european country
optional_result_slots: {}
- intent: request_ticket
description: the user specifies the service and the corresponding ticket is issued
original_task: POA02
required_slots:
- name: type_of_service
description: shipping a parcel, pay a bill
is_categorical: yes
possible_values:
- shipping_parcel
- pay_a_bill
optional_slots: {}
result_slots:
- name: ticket_number
description: The POA interfaces with the ticket issuing system, ticket number
associated with the service
optional_result_slots: {}
- intent: recharge_phone
description: the user wants to recharge a phone
original_task: POA04
required_slots:
- name: phone_number
description: phone number
is_categorical: no
possible_values:
- mobile or table phone number with country code
- name: import_payment
description: payment, amount of money
```

```
is_categorical: yes
possible_values:
- 10 €
- 20 €
- 30 €
- name: phone_provider
description: null
is_categorical: yes
possible_values:
- Poste mobile
- TIM Vodafone
- WindTre
- Fastweb
- TIM
- Vodafone
- Wind
- 3
- three
- CoopVoce
- Carrefour UNO Mobile
- Poste Mobile
- Daily Telecom
- A-mobile
- Conad INSIM
- MTV Mobile
- Fastweb Mobile
- Telepass Mobile
- BT Mobile
- Digitel Italia
- Digi.Mobil Italia
- Smart Pinoy
- Tiscali Mobile
- Noverca
- ERG Mobile
optional_slots: {}
result_slots:
- name: outcome_operation
description: if the data are corrects the POA ask to insert the card in the POS,
the user visualizes the outcome of the operation
optional_result_slots: {}
- intent: question_answering
description: The POA provide information about a generic financial product,
Postepay evolution, Linea Protezione Patrimonio
original_task: POA05, RA04, RA07
required_slots:
- name: question
description: the user asks a specific information about a product or a bill type
is_categorical: no
possible_values: null
optional_slots:
- name: type_of_bills
description: The type of bill/form for which the user wants to get information
is_categorial: yes
```

```
possible_values:
- PagoPA
- Empty Form
- MAV
- 896 (completed pre-printed payment slip)
- 674 (partially completed pre-printed payment slip)
result_slots:
- name: evidence
description: tha POA extract the information from the specific document
optional_result_slots:
- name: bill_form_name
description: name of the specific postal bill form
- name: bill_form_description
description: bill form characteristics
- name: bill_form_payment_procedure
description: informations for the filling form procedure and ticket
- intent: building_access
description: The visitor (who is not a POSTE employee) has to
participate to a meeting in a POSTE building. This interaction will
happens the day of meeting some time before of the meeting starting time.
All the information about the meeting room, start/end time, guest name,
etc... are available through the QRcode that the guest will receive in the
invitation for the meeting beforehand the meeting.
original_task: RA01
required_slots:
- name: guest_name
description: Name of the person who asks to enter the building
is_categorial: no
possible_values: null
- name: host_name
description: Name of the employee
is_categorical: no
possible_values: null
- name: alternative_host_name
description: Optional alternative host name
is_categorical: no
possible_values: null
- name: host_email
description: The host email corresponds to the account on the Microsoft Teams
server in order description to accommodate the guest confirmation call
is_categorical: no
possible_values: null
- name: alternative_host_email
description: The alternative host email corresponds to the account on the
Microsoft Teams server in order to accommodate the guest confirmation
call
is_categorical: no
possible_values: null
- name: meeting_date_and_time
description: Date and time of the meeting
is_categorical: null
possible_values: null
- name: meeting_room_identifier
```

```
description: Unique indentifier of the meeting room
is_categorical: no
possible_values: null
optional_slots: {}
result_slots:
- name: verification_call
description: The system will set a videocall to let the host to visually inspect
the guest and authorize the access
- name: Confirmation to open the turnstile
description: This is a signal toward the internal system that controls the
turnstile to let the guest to enter
- name: additional_safety_information
description: Under particular circumstances (e.g. COVID pandemic, area of the
building closed for the access etc…) before allowing the turnstile to
open the system must provide additional safety information to each guest
(as video or audio, or just as text in case of our dialogue collection)
optional_result_slots: {}
```

In the following Table 5 and Table 6, we show the list of emotions and user gestures that provide meaningful signals for this purpose. The list is basically a distilled version of the current research in gesture recognition by Noroozi et al. (2018). It is noteworthy that the lists are subject to change during the development of the project.

Table 5 - List of potential emotions that are collected in our dialogue data

| Emotion | Body Language |
|---|---|
| Anger | Body spread. Hands on hips or waist. Closed hands or clenched fists. Palm-down posture. Lift the right or left hand up. Finger point with right or left hand. Finger or hand shaky. Arms crossing. |
| Confusion | Tilt head. Narrow eyes. Furrowed brow. Shrug. |
| Curious | Lifting one eyebrow. Purse lips. Lick lips. Rub hands together. Chin-stroking gesture. |
| Disgust | Backing. Hands covering the neck. One hand on the mouth. One hand up. Hands close to the body. Body shifted. Orientation changed or moving to a side. Hands covering the head. Dismissive hand waving. |
| Fear | Noticeably high heartbeat-rate (visible on the neck). Legs and arms crossing and moving. Muscle tension: Hands or arms clenched, elbows dragged inward, bouncy movements, legs wrapped around objects. Breath held. Conservative body posture. Hyper-arousal body language. |
| Frustration | Eyes look around. Hands kept up, scratching one's head. Shrugged shoulders. |
| Happiness | Arms open. Arms move. Legs open. Legs parallel. Legs may be stretched apart. Feet pointing at something or someone of interest. Looking around. Eye contact relaxed and lengthened. |
| Neutral | No explicit expression |

| | |
|---|---|
| Sadness | Body dropped. Shrunk body. Bowed shoulders. Body shifted. Trunk leaning forward. The face is covered with two hands. Self-touch (disbelief), body parts covered or arms around the body or shoulders. Body extended and hands over the head. Hands kept lower than their normal positions, hands closed or moving slowly. Two hands touching the head and moving slowly. One hand touching the neck. Hands closed together. Head bent. |
| Stressed | Tense face. Stroke/rub nape of neck. Clasp hands over head. Wring hands. Run hands through hair. Adjust cuffs. |
| Surprise | Abrupt backward movement. One hand or both hands moving toward the head. Moving one hand up. Both hands touching the head. One of the hands or both touching the face or mouth. Both hands over the head. One hand touching the face. Self-touch or both hands covering the cheeks or mouth. Head shaking. Body shift or backing. Wide eyes (shock). |

Table 6 – List of user gestures prior to or during the verbal communication

| User Gesture | Description |
|---|---|
| Walk away from the agent | A user walk away from the agent in the middle of a dialogue |
| Point to an object | User point to an object in the post office |
| Cover the camera of the agent | A user covers the camera of the agent so the agent cannot get vision signals |
| The user waves at the agent to check if the agent reacts | The user tries to attract the agent's attention through specific gestures of the hand and/or of the arm. |
| The user indicates a location | While interacting with the agent, the user might indicate a location with their arm, hand, fingers, or gaze to clearly indicate a location, a target, or a position. |
| The user approaches the agent | The user approaches the agent while looking at it, showing interest in interacting with it. Vice versa, the user might be in proximity of the agent but looks and moves elsewhere (in this other case the user is not interested in interacting with the agent). |
| The user randomly looks around | The user looks confused while the interaction is in progress. The user probably needs help but did not notice the agent. |
| The user is looking for something | The user basically looks for something, e.g., in their pocket to find the ID card, and the agent needs to wait for him, or help. |
| The user handles an object | The user might pick up a parcel or bring a document, showing the intention to interact with the agent. |
| Walk away from the agent | A user walks away from the agent in the middle of a dialogue |

Table 7 – List of potential physical interactions from SERMAS agents to users

| Physical interactions | Description |
|---|---|
| Point to an object | Point to an object such as a shelf in the post office to show where the user can buy a brief? |
| Point to a direction | Show the direction to a room in case of RA |
| Navigation instruction | Receive a room with location from verbal communication and show the user how to go to the room |
| Handle a parcel | Raise the arms and hands up to allow the user to put the parcel on the hands |
| Show availability of interaction | The robot moves in the direction of the user, or switch on the screen, showing availability to interact with the user |
| Move smoothly | The robot approaches the user in a smooth way, showing comfort to the user |
| Keep a socially accepted distance | While executing a collaboration task with the user, the robot maintains a socially accepted distance from the user |

These possible physical interactions from the agent to a user are demonstrated in Table 7. Similar to the lists of emotions and user gestures, the list of interactions is subject to change during the development of the SERMAS project.

We note that the non-verbal communication signals collected at this phase are textual description (categories). We plan to extend and enrich the collected textual dialogue data by visual or sensing signals of corresponding emotions and body gestures. So that the final data is multimodality.

Table 8 – List of potential error types in the responses generated by the agent

| Level | Error Type | Description | Collect Additional Data |
|---|---|---|---|
| Response | E1 – Ignore Question | The agent utterance ignores the user's question. | The user's original question |
| | E2 – Ignore Request | The agent utterance ignores the user's request to do something. | The user's original request |
| | E3 – Ignore Expectation | The agent utterance does not fulfill the user's expectation. | The user's expectation |
| | E4 – Slot Error | The agent utterance suggests that the system did not get the slots right. | No additional data (the correct slot should be attached to the previous user's utterance) |
| | E5 – Factually Incorrect | The agent utterance contains information that is factually incorrect. | The wrong information + the |

| | | | |
|---|---|---|---|
| | | | correction / response alternative |
| Context | E6 – Topic Transition Error | The agent utterance transitions to another / a previous topic without reasonable explanation. | No additional data |
| | E7 – Conversationality | The agent utterance indicates that the system lost track, e.g., it repeats previous responses (without asking for missing information) or contradicts itself. | No additional data |
| | E8 – Unclear Intention | The agent utterance suggests that the user's intent was not successfully conveyed. | No additional data (the correct intent should be attached to the previous user's utterance) |
| Society | E9 – Lack of Sociality | The agent utterance lacks consideration of social standards, e.g., greetings, is toxic or disrespectful. | The disrespectful phrase + an response alternative |
| | E10 – Lack of Common Sense | The information in the agent utterance opposes the opinion of the majority. | Like E5 |

We will collect annotations for potential error types in the agent's utterances (response errors) and the potential feedback given by a user (user feedback). Table 8 presents the list of error types that could be annotated by the user after each response generated by the agent, for feedback collection. They are divided into three different granularity levels: response, context, and society.
Response-level errors describe situations in which the agent's response is not related to the upon user utterance. The causes include that the agent incorrectly recognizes the intent/slots, hallucinates content, or could not achieve the user's expectation.
Context-level errors are those related to the dialogue structure. Example situations are when the agent might lose track of the dialogue and repeats itself.
Society-level errors mostly describe situations in which the agent utterances contain harmful or disrespectful language. We note that society-level errors are highly subjective and hard to meet agreement.
We collect additional data for some of the response-level and the society-level error types. If an annotator marks a system utterance as ignoring expectation (E3; response-level), we ask for a free-text description of what was expected. Similarly, if the system utterance contains something factually incorrect (E5; response-level), we ask to provide the factually incorrect part of the system utterances and a brief free-text correction or alternative response. If an annotator marks a system utterance with a society-level error (E9 or E10), we ask to provide the

disrespectful/harmful phrase (E9) or a description of what was wrong (E10), along with an alternative response (E9, E10) or correction (E10).

Table 9 – List of potential user feedback

| Feedback Type | Description |
|---|---|
| FT1 | The user ignores the error and continues the dialogue. |
| FT2 | The user repeats or rephrases his/her concern. |
| FT3 | The user makes the system aware of the error and provides a correction. |
| FT4 | The user makes the system aware of the error without providing a correction. |
| FT5 | The user asks for clarification. |

We will also annotate user feedback types that determine whether a follow-up utterance by the user, after an error is raised, is the correction or a clarification question. We propose the feedback types in Table 9.

We note that the error types and feedback types are to be collected in the future. By the time of the submission, we mainly collected the human-human dialogues for evaluation.

# 4. Dataset Collection Process

We first present the plan for data collection. We then describe the recruitment of participants (the annotators), their tasks in the data collection and the annotation guideline. We then discuss the current collection strategy and plan. Finally, we summarise the current progress of the data collection.

## 4.1. Data Collection Plan

We aim at collecting:

- human-human dialogues: stimulating the dialogues between a human and an agent
- human-agent dialogues: dialogues between a human and an agent

The human-human dialogues will be used as the testing set for evaluating the generalizability of the dialogue agent. This is motivated by the facts that human interactions contain more surprising factors and language variety than when a human interacts with an automatic agent. Meanwhile, we collect human-agent dialogues to facilitate the training and intermediate evaluation of SERMAS dialogue agents with a focus on continual learning from user feedback and non-verbal communication signals. This fully aligns with the general goals of the SERMAS project, in which we emphasise on social acceptance and user experience.

## 4.2. Annotator Recruitment

### 4.2.1. Participants

In this section, we compare the pros and cons of four recruitment strategies with a focus on resource estimation and expected data quality (Table 10).

Table 10 – List of recruitment strategies, their pros and cons

| **Recruitment** | **Cost Estimation** | |
|---|---|---|
| | **Pros** | **Cons** |
| **Students** | • (Easy) recruitment<br>• (Expected) High data quality<br>• Contract binding → commitment | • Simulated dialogues → low variants<br>• bureaucratic overhead (contracts, extension) |
| **Crowdsourcing (Contractors)** | • Diverse background/demographics<br>• Easy to recruit participants<br>• Manage the annotator pool and payments | • Require noise control, may encounter scammers<br>• Fee for crowdsourcing platforms<br>• Not many crowdsourcing platforms support customized tools |

| | | |
|---|---|---|
| | | (Prolific matches our needs) |
| **User study (Volunteers)** | • Diverse background/demographics<br>• Easy to recruit participants, no contract binding<br>• Costs determined by us | • The payment process is unclear now (Wiki page) → we plan based on our assumption<br>• Participant management – 2 need to join at the same time for a dialogue<br>• No contract binding → no commitment |
| **Students and Crowdsourcing/User study** | • Good for time estimation and quality control (can address some issues after first round such as bias, descriptions)<br>• Diversity of dialogues | • Might take longer |

Table 11 – Overview of the total scores for each annotator recruitment strategy

| **Score -1 0 1** | **Total Score** | **Resource Estimation** | | | **Data Quality** | | |
|---|---|---|---|---|---|---|---|
| | | Time eff. | Cost eff. | Less management | Dial. diversity | Natural dial. | Less noise |
| Students | 2 | 1 | 0 | 1 | -1 | 0 | 1 |
| Crowdsourcing (Contractors) | 2 | 1 | -1 | 1 | 1 | 1 | -1 |
| **User study (Volunteers)** | **3** | **0** | **1** | **-1** | **1** | **1** | **1** |
| Students + Crowdsourcing/User study | 3 | 0 | 1 | -1 | 1 | 1 | 1 |

We summarize the pros and cons of resource estimation and data quality and produce the following scoring table to facilitate our recruitment decision in Table 11. The scores are selected from the set of –1, 0 and 1. We give –1 when the strategy is negative with respect to the evaluation measurement, 0 for neutral and 1 for positive. Based on this extensive analysis, we recommend the recruitment of students and crowdsource workers for our data collection. Although this two-participant recruitment process significantly increases our administrative overhead, we hypothesize that participant diversity can bring us more diverse dialogues with higher variation. Furthermore, we assume that dialogues collected with student participants are of higher quality in comparison to crowdsource workers.

At the current phase, we only collect the data from university students, who are currently attending bachelor or master programmes. The second data collection round for human-agent dialogues with feedback annotations will be conducted after the submission of the deliverable. Since our human-human dialogues will be used for main evaluation, the collected data at this phase will be used mostly as the validation and testing sets.

All recruited participants will be asked to sign a consent form which will ensure security and privacy of sensible data during the tasks.

## 4.3. Tasks of Annotators

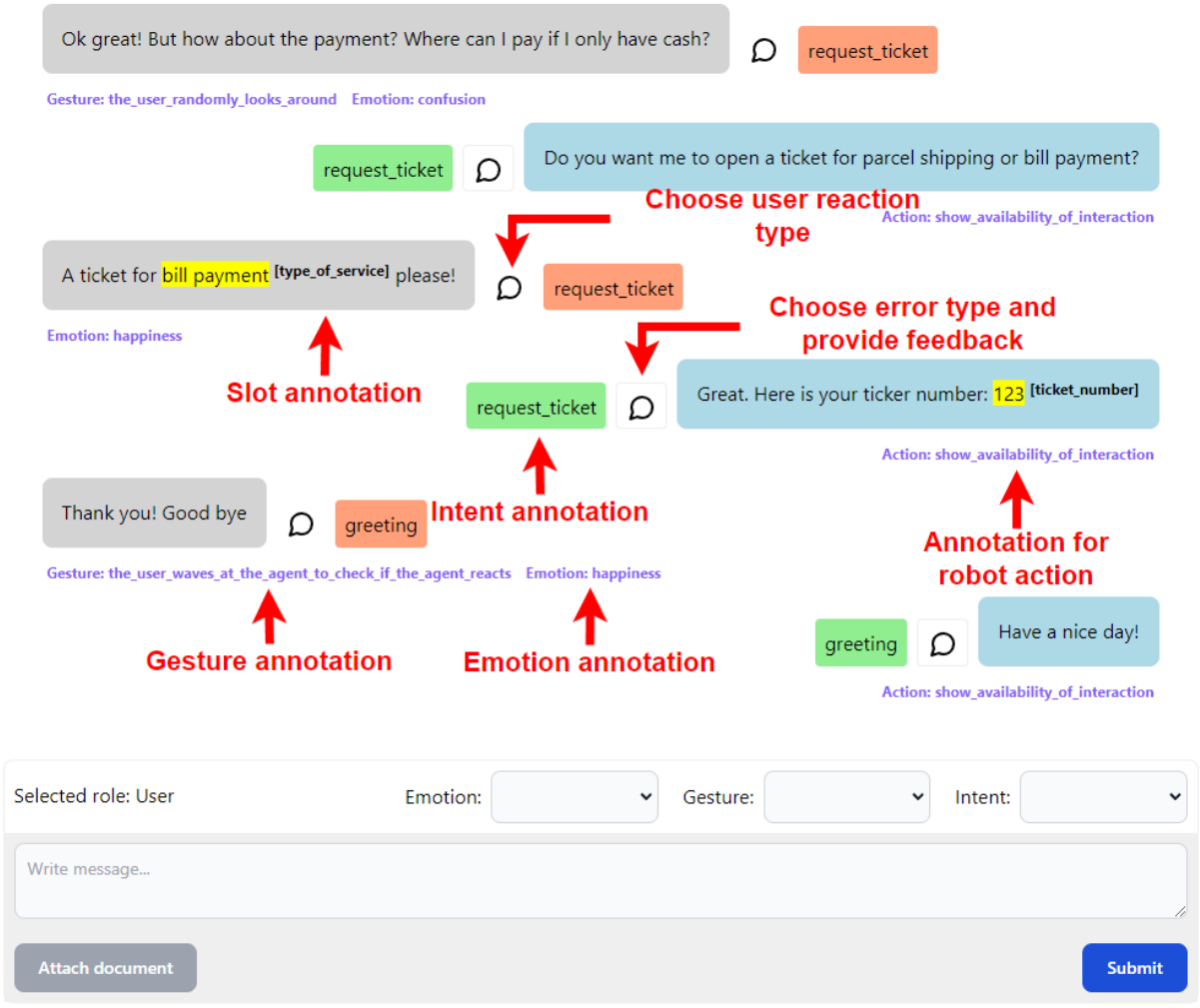


Figure 1. An example dialogue and the collected annotations

For the collection of the human-human dialogues, participants are provided with one of defined task descriptions and a list of predefined intents and slots. They are asked to talk to each other to fulfill their respective goals. Each human-human dialogue consists of a participant playing the role of a user and another subject acting as a dialogue agent. For the collection of the human-agent dialogues, the subjects are provided with the same task descriptions, but instead of messaging another human, they are messaging with a trained dialogue system to fulfill the task. Data collection will take place on a customized platform that will be provided

by SPXL (one of the project partners, described in the following section). illustrates an example dialogue collection with annotations.

For collecting the human-human dialogues, we form groups of two, e.g., two student assistants or two crowd workers, and assign them to sessions of 45 - 60 minutes. Each session is used to collect dialogues for one of the eight tasks.

We leave the human-agent dialogues for follow-up work as we are experimenting different baselines playing the role of the agent as well as automatic dialogue data generation ideas.

## 4.4. Annotation Guideline

At the beginning of each session, the subjects are provided with a task description that provides a description of the task, user classes, potential scenarios, an example dialogue, and information on slot values needed to fulfil the task. The dialogues are then collected in a role-play game manner. One subject plays the user. The other one plays the agent. The user is asked to randomly choose one of the user classes and to try to act accordingly based on personal experience.

In particular, the dialogues are mainly collected using the platform described in Section 5. The annotations are collected for each turn of a dialogue. Each turn consists of one user and one agent utterance. Utterances are always annotated with values for intents. Annotations for emotions and gestures in user utterances are optional (in case of emotions, we assume neutral as default). The same applies to annotations for actions in the agent utterances (we assume showing availability for interaction as default value). Slot values should only be provided if they provide values required to fulfil the task (on both sides). Annotations for error types and corrections (in case of agent utterances) or user response types (user utterances following an erroneous agent utterance) are only collected for human-bot dialogues.

## 4.5. Progress of the Annotation Collection

We collect a share of human-human dialogues for evaluation and testing.
We focus on human-agent dialogues as training data, as such data has a larger linguistic variety and is more likely to contain error situations in which users can provide helpful feedback. Errors and user feedback will only be collected in human-agent dialogues in the follow-up work.

# 5. Data Collection Platform

The scope of this section is to define the initial iteration on a messaging web application used to create datasets to train the WP5 dialogue model. We approach the development iteratively, trying to deliver in a few weeks the meaningful features to enable the task activities but also to review and adapt the scope to fit the natural evolution of the project. The messaging UI will allow one user to create dialogues simulating interactions between a user and a bot.

## 5.1. Architecture

The Data Collection Platform (DCP) is based on web-based technologies in the form of a SPA (Single Page Application). The DCP application is composed of a web frontend communicating with a backend RESTful API. The backend API offers interfaces to create and update messaging sessions and allow administrative users to download and manage the data available. Additionally, the backend exposes an MQTT socket (over WebSocket) to allow real-time exchanges between the acting users (or an agent). The application delegate authentication and authorization to a dedicated service offering interfaces and APIs to enforce user access and permission checks. Administrative users are allowed to manage users and review permissions.

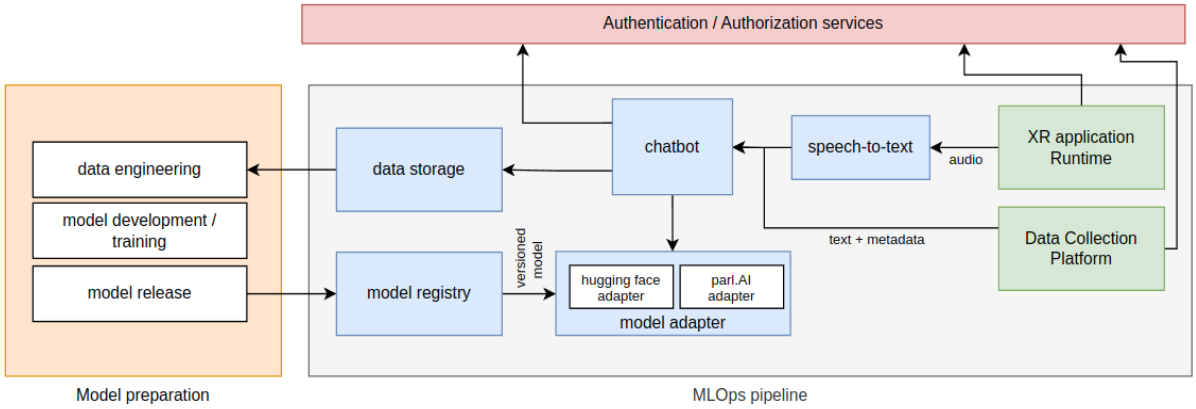


Figure 2. An example dialogue and the collected annotations

The proposed architecture encompasses the needs of the users involved. We organized the activities to clearly divide the scopes of activities to ensure each cross-functional team of each partner could operate autonomously.
For this extent, the architecture is designed to continuously inject data from the DCP but offers the same interfaces also to be reused by the Use Case in SERMAS to provide their interactions. The data collected and opportunely divided into sessions can be used as a base for data engineering, model development and training until an improved version is available.
The DCP backend offers adapters to support different language model providers (such as Parl.AI and Hugging Face) but exposes a unified API interface to ensure the underlying model can be changed or updated transparently for the users. The output of the model preparation steps provides refined and improved models that can be loaded in the corresponding adapters to offer improved dialogue capabilities

to the end-users. We leverage on a model registry to provide versions of the model based on the available releases.

### 5.1.1. Components

Figure 3 demonstrates the messaging UI in the annotation platform.

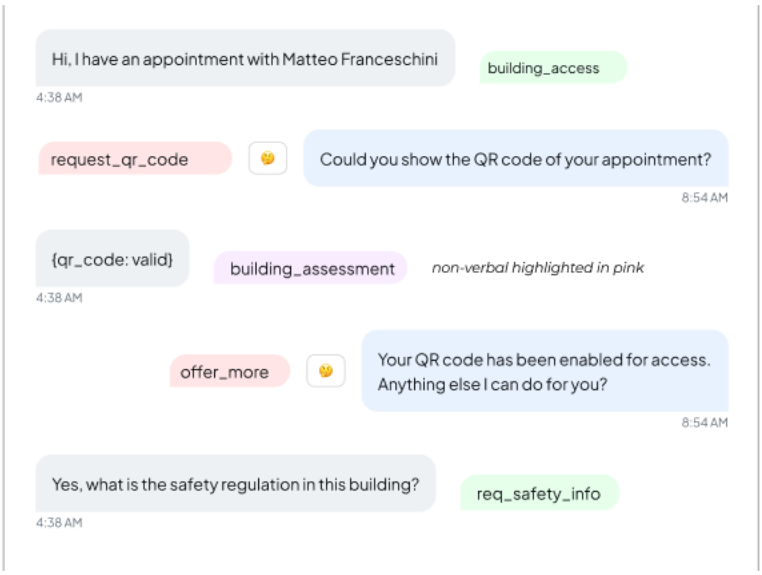


Figure 3. The messaging user interface

Data insertion leverage on a form with key information to be saved for the dialogue interactions, illustrated in Figure 4.

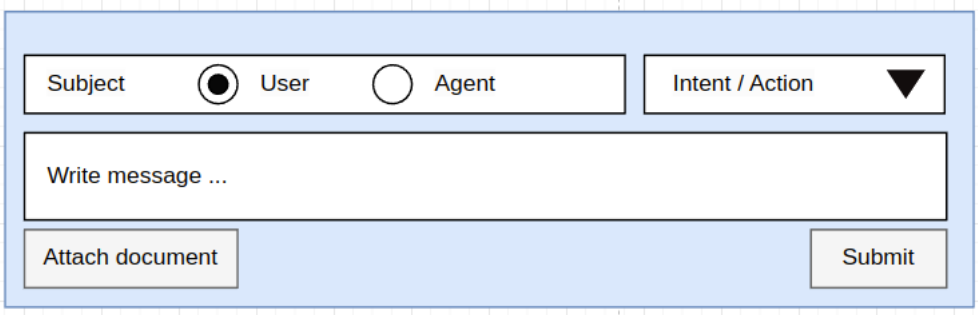


Figure 4. Simplified message input form

The first version of the DCP provides the following fields

- Subject: the user or the agent
- Intent/Action: identifying the intent of the dialogue
- Text: the message
- Attach document: Reference to one or more documents stored in a simple text format (e.g., plain text, markdown, xml) that can be taken from a public URL or directly uploaded.


Figure 5. User feedback collection UI

Figure 5 illustrates how error and feedback annotation collection could look like in the UI.

### 5.1.2.Storage and Data Exchange

The data model is designed as the follows:

```
Class UploadedDocument {
    source: “upload” | “url”
    reference: string
    documentId: string
    phrases: SlotSpan[]
}
class SlotSpan {
      span_type: string
      span: string
      span_character_start_position: string
}
class Feedback = {
   feedback: ‘satisfaction’ | ‘error’
   code: ‘e1’ | ‘e2’ | ‘e3’ | ‘e4’ | ‘e5’ | ‘e6’ | ‘e7’ | ’e8’ |
‘e9’ | ‘e10’
  source: string
  correction: string
  wrong: string
}
class DataRecord {
    subject: 'agent' | 'user'
    intent: string
    text: string
    slots: Map<string, string>
    feedbacks?: Feedback[]
    attachments: Attachments[]
    timestamp: Date
}

class Session {
     groupId: string
    sessionId: string
    authorId: string
    created_at: Date
    modified_at: Date
    records: DataRecord[]
}

class Attachments {
    source: string;
    reference: string;
    documentId: string;
    phrases: SlotSpan[];
}
```

Here is an example of an annotated dialogue session. When the user writes “Hi, I have an appointment with Matteo Franceschini” and the intent chosen from a list below is “building_access” the following data structure is created:

```
{
   groupId: “groupid_session”
    sessionId: “guid_session”
    created_at: 2022/12/19 04:32 pm UTC
    modified_at: 2022/12/19 04:32 pm UTC
    records: [
  {
    subject: ‘user’
    intent: ‘building_access”
    feedbacks: []
    text: “Hi, I have an appointment with Matteo Franceschini,
appointment with Mr. Rossi”
    slots: [{
      ‘span_type’: “employee_name”,
      ‘span’: “Matteo Franceschini”,
      ‘span_character_start_position’: 15
    }]
    attachments: null
    timestamp: 2022/12/19 04:32 pm UTC,
         emotions: ‘happiness’,
         gestures: ‘walk_away’,
         actions: “”
  },
  {
    subject: ‘agent’
    intent: ‘adk_qr_code’
    feedbacks: []
    text: “Could you show the qr_code of the appointment?”
    attachments: [
       {
         source: “upload”
         reference: “procedureABC.pdf”
         documentId: "<doc guid>",
         phrases: [{
          ‘span_type’: “span_type_1”,
          ‘span’: “an reference phrase”,
          ‘span_character_start_position’: 15
         },
         {
          ‘span_type’: “span_type_2”,
          ‘span’: “another reference phrase”,
          ‘span_character_start_position’: 42
         }]
       },
       {
         source: “url”
         reference: “https://example.com/”
         documentId: "<doc guid>"
                      phrases: []
       },
    ]
    timestamp: 2022/12/19 04:32 pm UTC,
          emotions: ‘‘,
```

```
            gestures: '',
            actions: "handle_parcel"
    }]
}
```

## 5.2. The Agent – the Dialogue Model

### 5.2.1. Training Settings

We build our baseline, the pre-trained language model – Flan-T5 (Chung et al. 2022) on the Transformers library (Wolf et al. 2019). The model is trained on instruction objective for text generation, which has been shown to generate fluent text and can be adapted to dialogue response generation. We fine-tune the model on FITs dataset (Feedback for Interactive Talk & Search; Xu et al., 2022), which is a dialogue dataset annotated with feedback in OpenDial.

### 5.2.2. Serving of the Pre-trained Dialogue Model

The pretrained model is loaded in the corresponding runtime and its interface exposed over a TCP socket (WebSocket) to enable asynchronous interactions between the user(s) and the agent model.

## 5.3. Functionality

After login, annotators can select or create a session that will contain the exchanges with another human annotator (or an agent in a second phase). The annotators insert in a messaging interface their interaction and then wait for the counterpart to provide a follow up. Annotators can select part of the text and indicate the meaning (slot) a specific word or sentence has (e.g., a full name, an address, or a functional world for the context, like "parcel"). Additionally, annotators can upload whole documents in text format and select the meaningful parts that will be stored as contextual reference for a specific interaction. The administrators can then download the sessions in JSON format to be used as training data for the model.
Once the model is stable, we will integrate the interaction between the newly trained agent model as one of the acting users in the data collection process. This will help in evaluating and providing feedback about the model's effectiveness and create meaningful datasets with corrections to further improve the behaviours of the agent.

## 5.4. Security

The DCP user interface is accessible over HTTPS ensuring data communication is encrypted. The storage of data is delegated to the Google Cloud Storage system that follows its own procedures to ensure compliance in the storage of data. The data collected has no personal data and the annotators' accounts are anonymized, lowering the overall requirements of security in this phase of the project.

# 6. The Collected Data

We report the collected data up to the deadline of the deliverable. At this phase, we only collect human-human dialogue data from students. We leave human-agent data for follow-up work as well as feedback collection. The enrichment with non-verbal communication such as images or raw sensing signals will be done as follow-up work.

## 6.1. Metadata

Table 12 – Metadata of the collected dataset

| Name | Description |
|---|---|
| Data set name | MuFeeTo (subject to change) |
| Format | Json |
| Purpose | WP5 – for POA and RA pilots |
| Storage | The data is currently stored at Google Cloud Storage. The data will be publicly shared on github along with the publication. |
| Utility | The data will be publised as a conference paper and in future scientific publications. |
| Specific sets | V0.1 (30/03/2023) |
| Description | • V0.1 contains only textual data<br>• Each instance consists of a dialogue between a human user and an agent that tries to support the user on some particular tasks. Note that in this version, both user and agent roles are played by our participants.<br>• The current version does not contain data that might be offensive, insulting, threatening, or causing anxiety. |
| Owners/Authors | UKP - TUDa |
| License | V0.1 - Private (as the data is not ready for publication on 30/03/2023) |

We recruited eight students from Technical University of Darmstadt with short-term hour-based contracts. Table 12 shows the metadata of our collected data, following the metadata reporting format described in D1.2.

## 6.2. Data Statistics

Table 13 – Collected human-human dialogues for the current dataset (v0.1)

| Collected Data | V0.1 |
|---|---|
| *Dialogue* | |
| # tasks | 8 |

| | |
|---|---|
| # unique intents | 6 |
| # fine-grained slots | 65 |
| # dialogues | 314 |
| # dialogues / task | 32 – 44 |
| Avg. dialogue length (utterances) | 12.89 |
| *Utterance* | |
| # utterances | 4,036 |
| # agent utterances | 2,061 |
| Avg. utterance length | 20.41 |
| # TOD utterances | 1,215 |
| # DocDial utterances | 1,580 |
| # OpenDial utterances | 1,241 |

Overall, we collected 4,036 utterances in 314 dialogues (sessions). The data is not split as we aim at using this only for testing. We reported the general statistics of the dataset in Table 13.

Table 14 – Statistics per task of the collected human-human dialogues (v0.1)

| Collected Data (v0.1) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | *Total* | *ba* | *pc* | *rp* | *qag* | *qahea* | *qafi* | *qaher* | *qap* |
| *Dialogue* | | | | | | | | | |
| # fine-grained slots | *65* | 13 | 23 | 32 | 9 | 9 | 16 | 9 | 9 |
| # dialogues | *314* | 40 | 35 | 41 | 38 | 32 | 44 | 41 | 43 |
| Avg. dialogue length (utts) | *12.89* | 13.15 | 14.23 | 14.8 | 11.37 | 11.88 | 11.52 | 13.12 | 12.74 |
| *Utterance* | | | | | | | | | |
| # utterances | *4,036* | 526 | 498 | 607 | 432 | 380 | 507 | 538 | 548 |
| # agent utterances | *2,061* | 268 | 252 | 307 | 222 | 194 | 261 | 275 | 282 |
| Avg. utterance length | *20.41* | 12.73 | 15.4 | 11.79 | 21.35 | 31.06 | 21.93 | 26.56 | 26.29 |

We present the detailed statistics of each task in Table 14. For brevity, we refer each task with their short forms in the table:

- *ba*: building access
- *pc*: parcel choice and request ticket
- *rp*: recharge phone and request ticket
- *qag*: QA about general products
- *qahea*: QA about health insurance
- *qafi*: QA about financial insurance
- *qaher*: QA about heritage insurance
- *qap*: QA about pet insurance.

For the details of the intents and their relevant slots, we refer the readers to Table 4. We plan to publish the data as a conference paper, which will include

follow-up work with feedback annotations. Thus, the data can only be made available upon request before publication.

## 6.3. Data Privacy

We will collect anonymous annotator information such as location, gender, and language distribution. The statistics of the author profile are demonstrated in Section 6.1. Since we only collect dialogues surrounding public services provided by POSTE without requiring personal private information from the annotators/workers, we consider that there is no privacy issue in the collection process. Personal information of the annotators is thus not revealed in the collected data. The documents from POSTE for QA are all publicly available on their website or brochures/leaflets. Hence, we see no privacy issues of these documents. Data collected using dialogue systems in human-agent scenarios will be verified and curated. Dialogs that may leak private information will be filtered out based on user interaction data.

We aim to publish the dataset in a conference paper, which is currently in preparation, and possibly in further scientific papers.

# 7. Findings and Lesson Learned

**Annotation process**

One may think that having both human annotators to join the same dialogue session and talk to each other would be easy. However, at the beginning of our human-human dialogue collection, we face several issues with syncing and misunderstanding between workers and of the data collection instruction. These issues include:

- Diversity of the collected dialogue data
    - Self-dialogues in which an annotator plays both roles: the user and the agent are less diverse than synchronous messaging between participants.
- Real-time human-human dialogues may create extra difficulties due to human nature of not immediately response.
- Participants without annotation experience are likely to be confused by the terminology that we use as well as the annotation process for slot. Thanks to the friendly design of our platform, the participants can pick up the idea at most after one dialogue.

**Annotation Platform**

- Some programming logics are not trivial without pilot study. A back-and-forth update of the software to reveal and resolve some logical issues is necessary for adapting the platform to different use cases.
- In general, the platform allows researchers to collect dialogue data in a nutshell.

# 8. Conclusions and Next Steps

In this deliverable, we have analysed the requirements of data collection for the project and the pilots. We also described the data collection process and platform. We reported the statistics of the data collected up to the deadline of the deliverable. We continue to collect as planned in the future. We also plan on aligning the collected non-verbal communication textual categories to existing visual datasets on emotions and body gestures or sensing signals in WP4, which can turn the data into multimodality. Further details of the alignment will be discussed with other partners.

# Reference

Alina Roitberg, Alexander Perzylo, Nikhil Somani, Manuel Giuliani, Markus Rickert, and Alois Knoll. 2014. Human activity recognition in the context of industrial human-robot interaction. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA)*.

Fatemeh Noroozi, Ciprian Adrian Corneanu, Dorota Kamińska, Tomasz Sapiński, Sergio Escalera, and Gholamreza Anbarjafari. Survey on emotional body gesture recognition. 2018. *IEEE transactions on affective computing 12, no. 2 (2018): 505-523*.

Chung, Hyung Won, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jing Xu, Megan Ung, Mojtaba Komeili, Kushal Arora, Y-Lan Boureau, and Jason Weston. 2022. Learning new skills after deployment: Improving open-domain internet-driven dialogue with human feedback. *arXiv preprint arXiv:2208.03270*.