applied sciences

MDPI

Article

KIDE4I: A Generic Semantics-Based Task-Oriented Dialogue System for Human-Machine Interaction in Industry 5.0

Cristina Aceta [1,*], Izaskun Fernández [1] and Aitor Soroa [2]

[1] TEKNIKER, Basque Research and Technology Alliance (BRTA), 20600 Eibar, Spain; izaskun.fernandez@tekniker.es
[2] HiTZ Center—IXA, University of the Basque Country UPV/EHU, 20018 Donostia, Spain; a.soroa@ehu.eus
[*] Correspondence: cristina.aceta@tekniker.es

Abstract: In Industry 5.0, human workers and their wellbeing are placed at the centre of the production process. In this context, task-oriented dialogue systems allow workers to delegate simple tasks to industrial assets while working on other, more complex ones. The possibility of naturally interacting with these systems reduces the cognitive demand to use them and triggers acceptation. Most modern solutions, however, do not allow a natural communication, and modern techniques to obtain such systems require large amounts of data to be trained, which is scarce in these scenarios. To overcome these challenges, this paper presents KIDE4I (Knowledge-drIven Dialogue framEwork for Industry), a semantic-based task-oriented dialogue system framework for industry that allows workers to naturally interact with industrial systems, is easy to adapt to new scenarios and does not require great amounts of data to be constructed. This work also reports the process to adapt KIDE4I to new scenarios. To validate and evaluate KIDE4I, it has been adapted to four use cases that are relevant to industrial scenarios following the described methodology, and two of them have been evaluated through two user studies. The system has been considered as accurate, useful, efficient, not demanding cognitively, flexible and fast. Furthermore, subjects view the system as a tool to improve their productivity and security while carrying out their tasks.

Keywords: task-oriented dialogue systems; human-machine interaction; Industry 5.0; semantics; natural language processing; collaborative robotics

check for updates

Citation: Aceta, C.; Fernández, I.; Soroa, A. KIDE4I: A Generic Semantics-Based Task-Oriented Dialogue System for Human-Machine Interaction in Industry 5.0. *Appl. Sci.* **2022**, *12*, 1192. https://doi.org/10.3390/app12031192

Academic Editors: Fabio Tango, Frederik Naujoks and Andrea Castellano

Received: 30 December 2021
Accepted: 21 January 2022
Published: 24 January 2022

1. Introduction

Recent technological advances in last decades have caused a great revolution in industrial settings. Is that so, that terms such as Industry 4.0—and even more recently, **Industry 5.0**—that define sustainable, advanced and human-centered industrial environments are now essential in modern industry.

Human workers are one of the fundamental pillars in this setting, and their wellbeing when performing their everyday tasks is of utmost importance. For this, the use of technologies such as artificial intelligence (AI) to facilitate their work in the production process is becoming widely extended nowadays. The implementation of more complex and innovative technologies in industrial scenarios, which have reduced the physical workload of workers, have, as a counterpart, an increase of the cognitive load so as to control and manage such technologies [1]. In this sense, workers interact with a wide range of systems at a daily basis, such as intelligent information systems or advanced collaborative robots, and it is key to facilitate this interaction so as to guarantee optimal work conditions. For this, task-oriented dialogue systems are powerful technologies that allow workers to perform multiple tasks at once by delegating simpler tasks through voice commands. Furthermore, the use of these technologies are conceived so as to not affect the quality of workers' tasks, as they only require a simple interaction for the target system to function, with minimal impact in their cognitive demand.

The possibility of communicating to industrial systems through natural language is highly encouraged since it triggers acceptance from humans [2]. Furthermore, it reduces workers' mental stress, since they do not have to memorize specific words or sequences to interact with the systems. However, to develop task-oriented dialogue systems of these characteristics by using current state-of-the art technologies, such as deep learning (DL) techniques, is a difficult task. The main challenge regarding these technologies is that great amounts of data for training are needed, and currently available data is usually bound to specific domains and is also scarce [3], especially in industrial scenarios [4]. Thus, most dialogue solutions designed for industrial settings are highly specific for the application they are intended, so their capacity to be reused in other scenarios is very limited and are usually bound to expert manual work and high development time and costs [5]. These solutions usually also make use of static structures and rigid language and, thus, communication through natural language is quite restricted [6,7].

These remarks motivate the development of **KIDE4I** (Knowledge-drIven Dialogue framEwork for Industry), the **generic semantics-based task-oriented dialogue system framework** presented in this paper. This framework, with the objective of enabling workers to naturally interact with industrial systems, uses semantic technologies as its core component, and its design is generic enough to allow an easy adaptation to different industrial applications—such as collaborative tasks, guidance, information systems, assistance, etc.—without requiring great amounts of training data to be constructed. Furthermore, its architecture is designed so it is language-independent.

So as to obtain a command that can be executed by the target system from a natural user command, KIDE4I consists of four main modules, which will be detailed throughout this work: a **key element extraction** component, which extracts the relevant key elements from a user command—a **polarity interpreter**—which determines if a user command is a confirmation or a negation—a **semantic repository**—which models the domain and the information necessary to manage the dialogue process—with the TODO [8] as its core and, finally, the **dialogue manager**, which strongly depends on the semantic repository logic and inferences and makes use of the rest of components according to the necessities of the dialogue process. The development of each of these components reuses existing ontologies and resources and technologies from the natural language processing field, which considerably reduces their adaptation time and effort.

To prove KIDE4I's easy adaptability to different scenarios, it has been adapted to four different use cases that are relevant to industrial settings, for the Spanish language: a guide robot—with capabilities that are analogous to logistics robots—a bin-picking robot, a computerized maintenance management software (CMMS) and an assistant for procedure execution. The first two adaptations have been validated and evaluated with two user studies, which are also described and reported in this work.

The rest of the paper is structured as follows: Section 2 provides relevant related work. Section 3 presents KIDE4I and its architecture, providing descriptions for its four modules, and Section 4 describes the adaptation process when a new use case is needed, and describes the different industrial use cases for which KIDE4I has been adapted to. Section 5 describes the experimental setup for the user studies carried out to validate and evaluate the system, and provides evaluation results. Finally, Section 6 provides some discussion regarding the findings of this work and future work directions.

2. Related Work

Task-oriented dialogue systems are usually based on pipeline architectures with a specific set of components that aim to supply a series of functions: to interpret the user's command—natural language understanding—to manage the dialogue process—dialogue state tracker and dialogue policy—and to generate the responses to be presented to the user—natural language generation [5]. The tasks to be identified and executed are usually conceptualized in terms of frame representations, which consist on modelling tasks in terms of a set of slots to be filled with the information provided by the user. If any information is missing, the dialogue system will engage with the user to obtain all the necessary information for the requested task to be performed [5].

In this context, traditional approaches for task-oriented dialogue systems rely on rules and templates for natural language understanding and dialogue management [9–11]. Nevertheless, recent advances in these fields have allowed the use of machine-learning-based techniques, both traditional machine learning [12–14] and, more recently, deep learning [15–17], which need large amounts of training data to be developed as a counterpart to not having to manually define rules or templates.

However, the reality for industrial scenarios is different. First, training data is scarce in this domain, and the use of machine-learning approaches is still very limited. Although there are attempts to combine rules and machine-learning techniques [18,19], rule-based approaches are generally used in these scenarios due to their specific characteristics [5,11]. As a consequence, most task-oriented dialogue systems for industrial scenarios are heavily adapted to the task they have been designed for and cannot be reused in other contexts, and developing new ones for new use cases is bound to expert work and high time and costs [5].

Furthermore, interaction in industrial contexts is usually oriented to one-way communication, from human to robot, and the system does not interact with the user in case there are inconsistencies or missing information [20,21]. In most cases, this interaction is limited to specific commands [20] and, in general, the possibility of using natural language is restricted.

Nevertheless, interaction with industrial systems is being oriented towards a natural communication [21,22]. The authors in [21] present a system that makes use of existing predicate-argument resources (Propbank [23]) to map natural commands to logical representations (e.g., *put(piece, box)* for "Put the piece in the box"). The reuse of existing, comprehensive resources of these characteristics allow a high flexibility in the type of requests that can be directed towards the system. Furthermore, the domain is represented by making use of ontologies, which allow a detailed modelling of the scenario and reduce ambiguity between different agents (in this case, human and target system) [24]. Nevertheless, it has some limitations in terms of dialogue (it is only unidirectional at the moment) and the reported implementation does not include variants (i.e., synonyms) for the different terms involved in the interaction (e.g., objects). As for the use of ontologies in these technologies, most of the task-oriented dialogue systems in the literature rely on them for domain modelling [25], but the tendency to use them for dialogue management purposes is increasing. The dialogue system presented in [26], besides domain modelling, also makes use of ontologies to implement a very simple dialogue state tracking. In this line, the approach in [27], OntoVPA, aims to obtain a dialogue system that is fully managed by ontologies, in which there is a distinction between a domain ontology and a dialogue ontology, which is used to manage the dialogue, keep track of the state of the dialogue and to store and control the responses and requests to be presented to the user. In this approach, requests and responses are highly dependent on the use case, whereas KIDE4I is designed to be generic enough so as to not need modifications in terms of responses and requests for the user, as they are parametrized. In the case of [28], with a similar approach to [27], the ontology used (Convology) is intended for dialogue policy planning so as to optimize the best path to complete a dialogue by using AI. However, this approach is limited to health-related applications, whereas KIDE4I is designed to be used in a wide range of scenarios inside the industrial domain. Also, the response outcome of the dialogue in KIDE4I is modelled in the ontology and, therefore, more controlled, whereas this solution generates it depending on the dialogue process.

Finally, none of the ontologies for OntoVPA and [28] (Convology) are publically available, whereas the TODO ontology can be easily accessed [8] to encourage its reuse and interoperability.

3. KIDE4I: A Generic Semantic-Based Task-Oriented Dialogue System

The main goal of KIDE4I is to obtain commands that are executable by the target system, given voice instructions uttered by the user. When the instructions are not clear, or when key information required to fulfill the goal is missing, KIDE4I engages in a conversation with the user and asks for missing information. The following lines will describe in more detail the process to obtain a target-system-readable command from user information and the components at play in it.

First, the user will perform a voice command, which will be transcribed and sent to KIDE4I. At this point, KIDE4I's dialogue manager will interpret the command by first extracting its relevant key elements according to a set of rules implemented in the key extraction component. Once these key elements have been obtained, the information on the semantic repository is exploited so as to obtain, from these key elements, the action to be sent to the target system and the necessary elements needed for that action to execute successfully (arguments), in a target-system-readable format. After processing all the information in the command, the dialogue manager checks again with the semantic repository whether all the necessary information has been obtained. In some cases, there will be information that is missing or the system will need the user to confirm certain pieces of information, and the system will require a response from the user. In those cases, the semantic repository will provide the dialogue manager the request to present to the user and the type of information expected from that response (i.e., a piece of information or a confirmation—in the form of *yes*/*no* and equivalents). After receiving the user response, and depending on the type of information to process, the dialogue manager makes use of the key element extraction component, mentioned above, or the polarity interpreter, so as to interpret pieces of information or confirmations, respectively. Once the dialogue manager checks with the semantic repository that all the necessary information has been obtained, the command to be sent to the target system is generated and sent for its execution.

The architecture of the semantic dialogue system presented in this work and mentioned above can be seen in Figure 1. In it, the 4 main components of the dialogue system can be distinguished: the **key element extraction** component, the **polarity interpreter**, the **semantic repository** and, finally, the **dialogue manager**. These components have been developed so as to be language independent and, thus, no additional effort to adapt KIDE4I to different languages is necessary. Furthermore, each of these modules are designed to be dockerized for an easy and fast deployment in server machines. Ontology instantiations are to be made available through the RDF store Virtuoso [29], which allows to access and infer information from the instantiated data through the standard querying language for RDF SPARQL, used by KIDE4I.

The following sections will describe in more detail each of the components and their function.

3.1. Key Element Extraction

The main function of the key element extraction component, thoroughly described in [18], is to obtain the relevant key elements from a transcribed user voice command that conveys a piece of information. Furthermore, it has been designed so as to learn over time from new interactions by generating training data semiautomatically to implement a supervised key element extraction component in the future.

Figure 2 shows the architecture for the component. When the command arrives to the key element extraction component, first, its syntactic analysis is obtained through the linguistic tool Freeling [30], in its version 4.0.

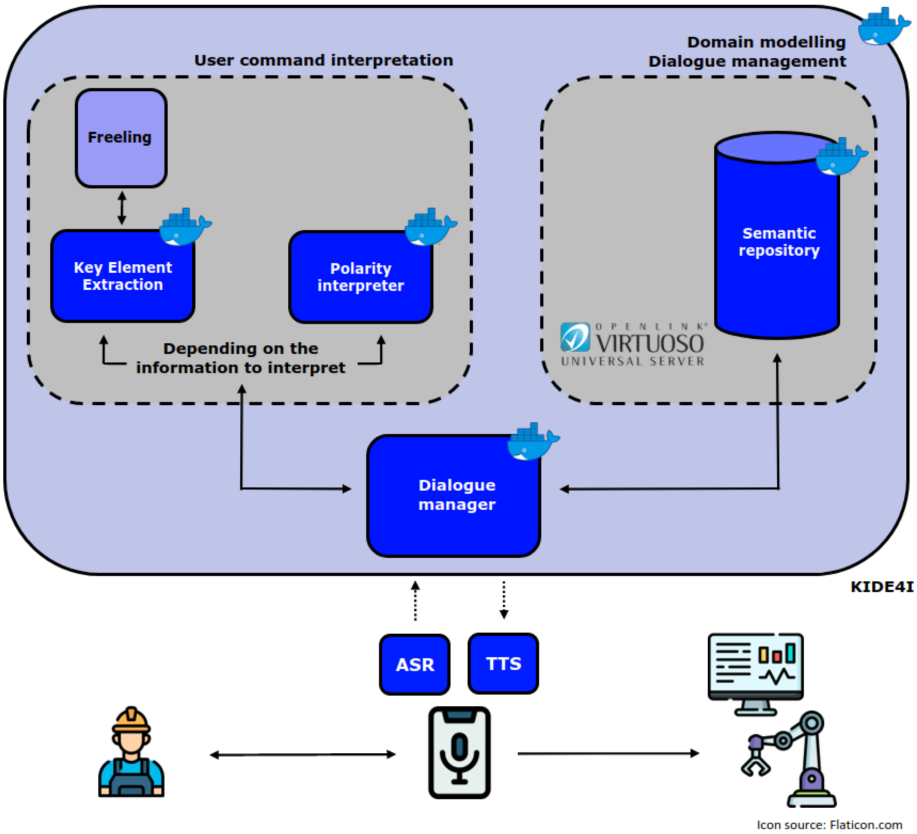


Figure 1. KIDE4I architecture.

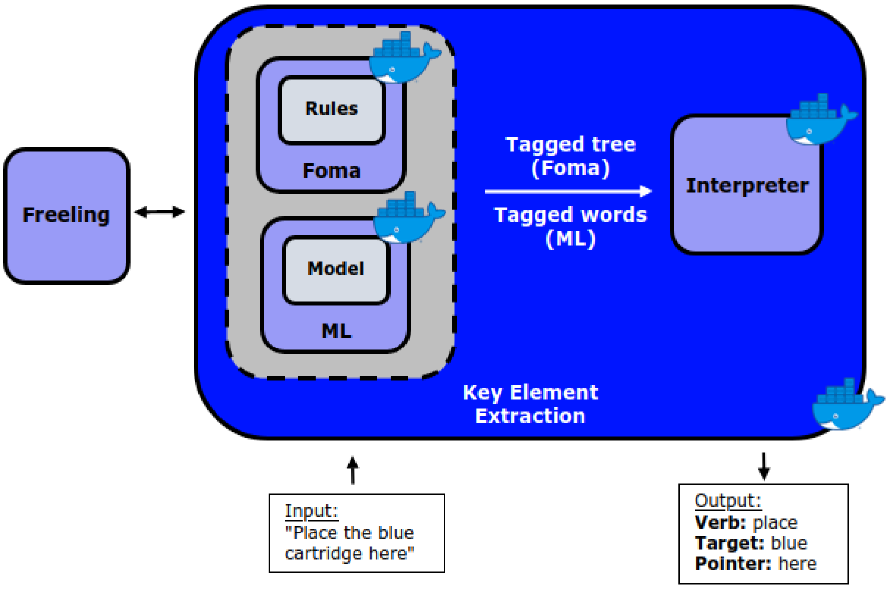


Figure 2. Architecture of KIDE4I's key element extraction component.

This syntactic representation of the user command serves as input to the next subcomponent, **Foma** [31]. Foma is a software that is able to convert regular expressions—defined in grammars—to finite-state automata, which is used in different natural language applications, such as morphological analysis. In this case, *definitions* and *rules* [18] need to be defined. On the one hand, definitions—most of which can be reused between use cases—model the different syntactic structures that are of interest for the use case (e.g., a *noun phrase* can appear in the syntactic tree as a noun—*container*—or as a noun + adjective—*blue container*). On the other hand, according to the necessities of the use case and considering the modelled definitions, a set of rules have been defined (e.g., *targets* appear as *noun phrases* in the use case at hand). These rules are used to delimit by tags the parts in the syntactic tree that correspond to the key elements in a given use case.

Since the syntactic tree does not include the words involved explicitly, but through indexes, a last step in processing is necessary to obtain key elements from user commands. Given a tagged syntactic tree, the **interpreter** subcomponent extracts the words that correspond to the indexes in the tree and classifies them according to the key elements defined for the use case.

The result of this key element extraction component is the set of key elements extracted from the user command, classified considering the necessities of the use case.

Once enough data has been obtained to train a supervised model, the Foma subcomponent is to be complemented with a machine-learning-based subcomponent that will tag the user command at word level, using BIO tags, considering the syntactic tree obtained from Freeling [18]. In this case, the interpreter will parse these word-BIO tag pairs to obtain the set of keywords for the command in the format described above.

3.2. Polarity Interpreter

In certain situations, the system needs the user to confirm specific pieces of information and it requests the user to provide a response that corresponds to an affirmation or a negation. Since one of KIDE4I's main characteristics is to process natural language commands, affirmations or negations can be provided in various forms other than typical *yes* or *no*. Thus, a component that determines the polarity of the response (that is, if the user responds positively or negatively to a system request), is necessary. This function is supplied by the polarity interpreter.

As far as the authors are concerned, the resources that determine if a string is equivalent to *yes* or *no* are scarce—even more in languages other than English—and in a very initial development stage. Considering this, the polarity interpreter makes use of sentiment analysis technologies—which are in a more advanced state—for the target language. For example, in the case of Spanish, which is the target language for the use cases reported in this work, this component implements the library *senti-py* [32] which, given a text, it provides a polarity score. Then, the command is classified as positive or negative according to a defined threshold, obtained through the exploration of several examples of constructions equivalent to confirmations and negations.

The output of this module consists on a boolean value that determines if the input has a positive (i.e., 1) or a negative polarity (i.e., 0).

Since this component is only dependent on the target language, it can be reused through use cases as long as the target language is the same.

3.3. Semantic Repository

The semantic repository stores all the information that allows the dialogue manager to function. It contains both knowledge at class level—commonly known as terminological box or *TBOX*—and the individuals that belong to those classes—assertional box or *ABOX* [33].

The core of this semantic repository—and, hence, the dialogue system—that corresponds to the *TBOX* in this framework is the Task-Oriented Dialogue management Ontology (TODO) [8]. This ontology consists of different modules that allow to model the dialogue management (TODODial) and the domain (TODODom) knowledge areas of the dialogue system.

Additionally, the *ABOX* stores, on the one hand, for the dialogue management dimension, (i) the requests and responses that the dialogue system can output to the user, (ii) the processing functions that the dialogue manager must perform given the key elements obtained from a user command and (iii) the implications of the different outputs of these functions. The advantage of this approach is that most instances and relations of this dimension can be reused through use cases and even languages, only requiring translations for requests and responses.

On the other hand, for the domain-related knowledge, the *ABOX* contains (i) the modelling of all the world elements of the domain and their relations, (ii) the different actions that can be performed by the target system, along with their arguments, (iii) the world elements that could belong to those arguments, (iv) the different variants to refer to both domain elements and actions in the target language and (v) the target-system-readable equivalents for world elements and actions. Finally, the *ABOX* is also the destination for the traces generated in each interaction with the dialogue system.

To sum up, the semantic repository determines the outputs the dialogue system will show to the user, the flow of the dialogue and allows the dialogue manager to interpret user commands to obtain a readable input for the target system. This means that the total control of the dialogue process depends on semantic-technology-based resources.

3.4. Dialogue Manager

The objective of the dialogue manager of the dialogue system framework presented in this work is to obtain, from a user command in text form that is received as input, a command that is readable for the target system.

This dialogue manager consists of two services: *init* and *userInput*. The first one generates a dialogue identifier and retrieves from the semantic repository all the necessary information to initiate a dialogue with the user, such as the first step of the dialogue process. Usually, this *init* service fetches the initial system request for the user (e.g., when the dialogue is modelled to initiate the dialogue with a greeting or a request for user input).

The *userInput* service is the target service for each user interaction. Given a user input, the dialogue manager relies on the knowledge in the semantic repository to determine of which type must be considering previous system output—either a *yes/no response* (i.e., *yes* or *no*) or a *content response* (i.e., a response that conveys a certain information). Depending on that knowledge, this component calls the polarity interpreter or the key element extraction modules to obtain an interpretation, respectively. Once an interpretation is obtained, the dialogue manager relies again on the semantic repository to determine the next function to execute or the action or world elements the user is referring to, and assert whether the information obtained is consistent or sufficient to obtain a readable command for the target system.

When the dialogue manager has checked with the semantic repository that all the information necessary to obtain a target-system-readable output is gathered, it generates the command for the target system in the corresponding format relying, again, on the information in the semantic repository.

4. Adaptation

So as to validate the dialogue system architecture presented in this work, it has been adapted to a series of use cases that are of relevance in industrial scenarios—a guide robot, a bin-picking robot, a computerized maintenance software (CMMS) and an assistant for procedure execution—which will be further described in the following sections.

The process of adaptation consists of 4 steps, basically based on the different components that are at play in the architecture of the dialogue system. It is important to remark that KIDE4I's components are designed to be language-independent and, thus, this same process applies to all languages:

1. Characterization of the use case. This preliminary step allows the developer to identify the necessities of the use case in order to be applied to the different modules of the dialogue system. This necessities include the type of interactions to be solved, the elements included in the domain, the key elements that refer to them and their possible syntactic structures, the target system's functionalities to be identified and the different situations (defined through frames [34]) that apply to each functionality in the use case.

2. Modelling of the key element extraction component. After having identified the key elements to be extracted, two main steps can be distinguished to obtain a functional key element extraction component:
   - Definition of Foma rules to delimit the structures that correspond to the previously defined key elements from the command's syntactic tree.
   - Adaptation of an interpreter that is able to obtain, from a given command and its tagged syntactic tree obtained from Foma, the set of relevant key elements to be used as input for the dialogue system.
3. Ontology modelling and instantiation. In this step, the necessary information to model the use case must be identified and instantiated into the TODO ontology. This step follows different phases, which are closely related to the different TODO modules:
   (a) Modelling and instantiation of the domain (TODODom). This phase is associated to two main blocks of knowledge, both related to the domain of the use case: world elements and action- and frame-related elements.
      - World elements (TODODW). Given that TODODW's classes are highly dependent on the use case [8], in this phase the domain elements are modelled and instantiated: objects, people, machines, spaces, etc., along with the relations that are relevant for the use case (e.g., a given workshop contains a given machine).
      - Frame- and action-related elements (TODODFA). The frames and related information required to successfully identify and process actions (e.g., arguments) are instantiated.

      For each of these blocks, the machine-readable information for the target system to perform such actions is also instantiated, along with the different words to refer to them in natural language when directing a command to the system (lexical units). These words are mostly obtained through automatic methods that rely on existing resources from the natural language processing field or database information and, when necessary, manually.

   (b) Modelling and instantiation of dialogue-management-related information (TODODM). This instantiation phase consists on the definition of the logic implications of the different outcomes of each interaction between the system and the user. In this stage, the responses and requests of the dialogue system are also defined. This information can be reused from other use cases and, if necessary, new elements must be modelled.
4. Adaptation of the source code of the dialogue manager. Although the source code is intended to be generic, it still needs minimal modifications that deal with the particularities of each use case, which are basically two: key elements processing (different use cases may have different configurations of key elements) and, if necessary, ontology queries to correctly interpret the commands directed to the dialogue system. These particularities are encapsulated in functions that have been designed to be easily adaptable. Further modifications are also needed when additional functionalities are needed, considering that the most typical ones are already defined.

The following sections will describe the adaptation process for each of the use cases previously mentioned.

4.1. Use Cases

As described throughout this work, KIDE4I is intended to be easily adapted to different use cases. In industrial scenarios, most typical interactions for workers are, on the one hand, with information systems to retrieve information about maintenance tasks or access specific information such as blueprints or technical manuals. On the other hand, workers also take part in collaborative tasks with robots, in which both work together towards completing an assignment [35]. In the context of this work, KIDE4I has been adapted to four different use cases that are relevant in current industrial setups considering the typical scenarios cited above: a guide robot—the characteristics of which are analogous to logistics robots—a bin-picking robot, a Computerized Maintenance Management Software (CMMS) and an assistant for procedure execution.

4.1.1. KIDE4Guide: Guide/Logistics Robot

The first use case consists on interaction with a guide robot: Teknibot (Figure 3). This robot, which has been presented in previous works [18], presents similar characteristics to logistics robots, and has the capability of moving from one point to another: given a user command, it is able to guide its target users to their destination of choice in a given environment and, additionally, is able to give information about specific objects or spaces. The target language of the use case is Spanish.

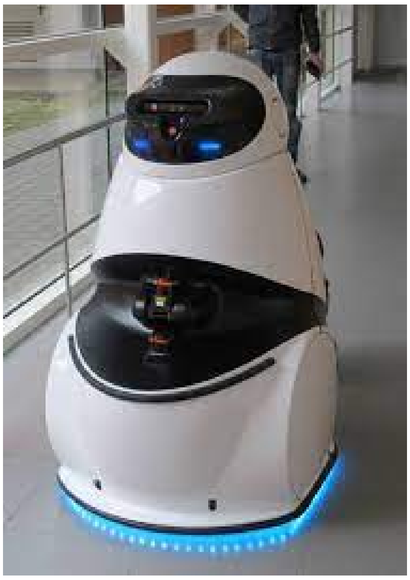


Figure 3. Teknibot is a guide robot that is able to guide its users to a given destination.

In this case, KIDE4I has been adapted to this use case so as the guide robot is able to guide around the research center Tekniker, providing guidance to its different laboratories, workshops, people, etc. and to give information about them.

The first two steps of the adaptation process can be observed in detail in [18], in which the key elements to detect and the Foma rules to obtain them were defined. In a nutshell, the relevant information to extract from commands for this use case are **actions** and **destinations** (which consist of a **target**, **preposition** and the target that depends on the preposition—the **complement**). Example (1) shows an instance of the previously mentioned key elements.

(1) a. Quiero $\textbf{ir}_{action}$ a una $[\textbf{sala}_{target}$ $\textbf{con}_{preposition}$ $\textbf{PC}_{complement}]_{destination}$
    b. I want to $\textbf{go}_{action}$ to a $[\textbf{meeting room}_{target}$ $\textbf{with}_{preposition}$ a $\textbf{PC}_{complement}]_{destination}$

For the domain-related part of the modelling and instantiation phase, the corresponding classes for the spaces, objects and people from Tekniker were modelled in TODODW. For this, terms from the GEO [36] and FOAF [37] ontologies were reused. Then, the instances of those classes were extracted from existing databases and instantiated automatically through ODBA mapping rules, and the relations between them modelled. The actions that could be performed by the system were also modelled and, for the frame-related information (frames, frame heads and related lexical units), the population strategy in [34] (which will be referred as *the population strategy* from now on), which makes use of multilingual existing language resources, was used. As for the machine-readable information for each element in the domain, it was modelled considering the requirements of the robot and its components. Finally, the different lexical units corresponding to the different domain elements that could not be obtained through the automatic methods mentioned above were obtained combining thesauri and expert knowledge.

On the other hand, the dialogue management information was modelled considering the expected outcomes of an interaction with a guide robot of these characteristics.

Tables 1 and 2 include information on the modelling and instantiation of the ontology for this use case, respectively. As it can be observed, more than 90% of the classes have been reused, and more than 70% of the instances have been obtained automatically.

In this case, this guide robot was used as the base use case to design KIDE4I and, thus, TODO, which is based on an initial conceptualization of KIDE4Guide. However, to preserve Table 1's consistency, TODO has been considered as reused for this use case, as new modifications in ontology modelling were needed in further development stages of KIDE4Guide. On another side, and considering this, other adaptations may make use of the rest of the work developed for this scenario (e.g., Foma rules or ontology instances).

Table 1. Classes for each KIDE4I adaptation: total and reused from other resources (TODO included).

	Dialogue		**Domain**		**All**	
	Total	**Reused**	**Total**	**Reused**	**Total**	**Reused**
KIDE4Guide	81	73 (90.1%)	43	43 (100%)	124	116 (93.5%)
KIDE4BinPicking	73	73 (100%)	40	36 (90%)	113	109 (96.5%)
KIDE4CMMS	73	73 (100%)	35	35 (100%)	108	108 (100%)
KIDE4Assistant	73	73 (100%)	171	169 (98.8%)	244	242 (99.2%)

Table 2. Instances for each KIDE4I adaptation: total and obtained automatically.

	Dialogue		**Domain**		**All**	
	Total	**Auto**	**Total**	**Auto**	**Total**	**Auto**
KIDE4Guide	110	75 (68.2%)	604	449 (74.3%)	714	524 (73.4%)
KIDE4BinPicking	75	75 (100%)	278	201 (72.3%)	353	276 (78.2%)
KIDE4CMMS	75	75 (100%)	150	99 (66%)	225	174 (77.3%)
KIDE4Assistant	75	75 (100%)	546	478 (87.5%)	621	548 (88.2%)

4.1.2. KIDE4BinPicking: Bin-Picking Robot

The second use case the dialogue system was adapted for is a bin-picking robot. This robot, which can be seen in Figure 4, is able to pick different printer cartridges from a table, identify their brand and color and classify them between two different containers according to the criteria established by the operator (that is, whether the cartridges of a specific color or brand should be placed in a container or another). It is also possible to interact with the system through gestures to convey the destination container or cartridge or to make the robot stop or continue. In this sense, gestures can be complementary to voice commands. As in the previous case, the target language for interaction is Spanish.

For this use case, the key elements to extract are **actions** and **targets**—which may correspond to **brands**, **colours** and **containers**—as it can be observed in Example (2). Furthermore, the key element extraction is also adapted to detect **pointers** that may imply the presence of a gesture referred to a container or cartridge, such as "here" or "this", as Examples (3) and (4) show.

(2) a. **Pon**$_{action}$ el **azul**$_{target-colour}$ en el contenedor **1**$_{target-container}$
b. **Put**$_{action}$ the **blue**$_{target-colour}$ one in container **1**$_{target-container}$

(3) a. **Pon**$_{action}$ **este**$_{pointer-cartridge}$ en el contenedor **1**$_{target-container}$
b. **Put**$_{action}$ **this**$_{pointer-cartridge}$ one in container **1**$_{target-container}$

(4) a. **Pon**$_{action}$ el **azul**$_{target-colour}$ **aquí**$_{pointer-container}$

b. **Put**$_{action}$ the **blue**$_{target-colour}$ one **here**$_{pointer-container}$

An initial analysis of the data generated for the guide use case has showed that the definitions for its key element extraction component could be reused, so only small modifications on definitions to include pointers and rule modelling were necessary to create the FOMA rules for key element extraction. For the interpreter, only minimal adaptations were required. This reuse of definitions has drastically reduced the time to obtain Foma rules which, added to the generality of the interpreter, has allowed to obtain a functional key element extraction component in a reasonable amount of time, reducing around a 90% of the work required.

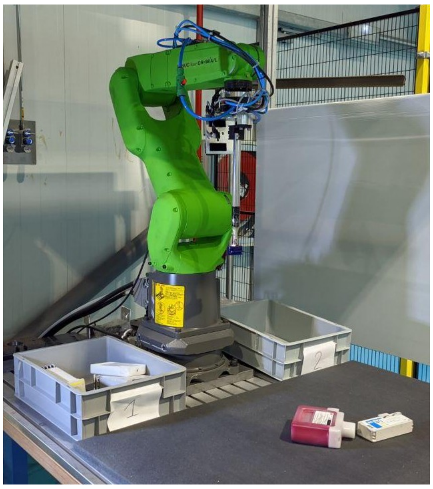


Figure 4. Bin-picking robot the dialogue system has been adapted for.

The modelling and instantiation of the world elements from the domain was performed in terms of the colours and brands that were recognized by the robot (cyan, magenta, black and yellow and Epson, Canon, HP and Brother, respectively) and the containers in the scenario (1 and 2). For cartridge colors, the Printer Vocabulary ontology [38] was reused, as well as the GEO ontology. As in the guide robot use case, actions were also modelled and frame-specific information was obtained by following the population strategy. Machine-readable information was modelled according to the robot's requirements and lexical units were obtained from linguistic resources (through thesauri and the strategy) and expert knowledge.

As Tables 1 and 2 show, a 96.5% of the classes have been reused, and a 78.2% of the instances have been obtained automatically, which means that the effort to model and instantiate the ontology has been highly reduced. Interestingly enough, the original dialogue management classes and instances were reused, although minor modifications were performed in the dialogue logic.

As it can be observed from the adaptation process for this robot, an important amount of data can be reused between use cases or obtained automatically, which validates the fact that this dialogue system framework is easily adaptable to other applications.

4.1.3. KIDE4CMMS: Information Systems for Maintenance Management

The third use case consists on interaction with a computerized maintenance management software (CMMS). By using this software, which is typically used to manage maintenance actions, users are able to access maintenance-related information, such as work orders or blueprints. The interaction target language is Spanish.

KIDE4I has been adapted so as to track work orders, request for blueprints, problem solving protocols or exploded views and check stock for a specific machine or machine component through natural language commands. Furthermore, users can also fill forms on the system's request.

So as to interpret each of the commands directed to the system, the key elements to extract are **actions**, **targets** and **items**. As Example (5) shows, *targets*, along with *actions*, determine the action to perform—in this case, to show a work order—whereas *items* are the action arguments—the work order identifier and the machine that work order is for.

(5) a. **Muéstrame**$_{action}$ la **orden de trabajo**$_{target}$ **85**$_{item}$ de la **fresadora**$_{item}$
b. **Show**$_{action}$ me the **work order**$_{target}$ **85**$_{item}$ for the **milling machine**$_{item}$

As in the previous cases, the guide use case's definitions were reused, and only 6 specific rules were defined for this use case. To cover the different key element tags, the interpreter was modified to include them.

For the domain modelling and instantiation, the available error codes—for the problem solving protocols—machines and components were modelled and instantiated, along with the IDs for work orders, which were defined as numerical patterns to be checked by the target system. Each action the system was able to perform was modelled, and the lexical units to identify targets and the rest of the elements of the scenario were modelled as in previous cases, both reusing existing lexical resources and linguistic knowledge. Following the CMMS' requirements, target-system-readable information was also modelled. Finally, action- and frame-related data was instantiated by following, again, the population strategy (more details on the use of this strategy in this use case can be found in [34]).

Finally, as Tables 1 and 2 show, all classes for this use case have been reused, and a 77.3% of the instances were obtained automatically. For dialogue, the original modelling and instances were totally reused.

4.1.4. KIDE4Assistant: Information Systems for Assistance

The fourth and last use case presented in this paper is an assistant for procedure execution, in the context of EKIN project [39]. Given a set of maintenance procedures, previously extracted from technical manuals, the system is able to guide the user through the processes described in them. The system has been designed for Spanish, and manuals are also in this language.

In this use case, procedures are structured in *methods*, *tasks* and *steps*. *Methods* determine different ways to perform the same procedure (e.g., in normal conditions or in a clean room); each *method* has a set of *tasks* (e.g., extract a battery, install a battery), and each *task* consists of a set of *steps* (e.g., disconnect the machine, open the lid). Given a procedure, the system requests the user to select the method to follow—if the procedure has more than one—Then, the system gives the description of the current method, task or step. Users are able to navigate through the different elements of the procedure by (i) requesting for the next or previous step or task (given the current step), (ii) to repeat the information that was just given by the system, (iii) to restart a method or task (i.e., start over again from the first step of the first task of the current method or to start over again from the first step of the current task, respectively), (iv) to obtain other related information such as the list of necessary tools to perform the procedure or (v) a more extensive description or (vi) additional information, in the form of text and images. Long texts and images are shown in a screen so users can easily follow the information provided by the system, whereas shorter texts are uttered.

Thus, the key elements to be identified by the target system are **actions** and **targets**, which correspond to the key word used to determine the action to perform (Example (6), for the *Show tool list* action) or the reference element of the action (Example (7)).

(6) a. **Muéstrame**$_{action}$ la lista de **herramientas**$_{target-determineAction}$
b. **Show**$_{action}$ me the **tool**$_{target-determineAction}$ list

(7) a. **Reinicia**$_{action}$ la **tarea**$_{target-reference}$
b. **Restart**$_{action}$ the **task**$_{target-reference}$

In this case, as the key elements to identify were common with the guide use case ones, both definitions, rules and interpreter scripts are also common.

The ontology design phase for this use case has two parts. On the one hand, 6 procedures—extracted from the manuals of 2 robotic arms and a controller, and formatted as JSON files—along with the relations between *procedures*, *methods*, *tasks* and *steps* are modelled, including sequential relations (e.g., *Step* 2 comes after *Step* 1 and before *Step* 3) are automatically instantiated into TODODW. For each method, their tool list is modelled, along with the tools in each list. For each structural element of the procedure (i.e., *methods*, *tasks* and *steps*), additional information and/or extended information is also automatically included. Furthermore, the elements that make reference to procedure parts—*procedures*, *methods*, *tasks*, *steps* and *tools*, which correspond to the key elements labelled as *targets*—were also modelled. So as to be able to instantiate this information, the VAR ontology [40]—which is intended as a "workplace digital twin" [40] by representing workplaces, processes and workers—as been reused.

On the other hand, the rest of the domain information is modelled and instantiated as in the previously described use cases, using the same methods.

Tables 1 and 2 show that nearly all the classes for this adaptation were reused (a 99.2%), and an 88.2% of the instances were obtained through automatic methods or reused from other use cases. Although all of the dialogue information was reused, minimal modifications on the dialogue's logic were needed to cover the necessities of the use case.

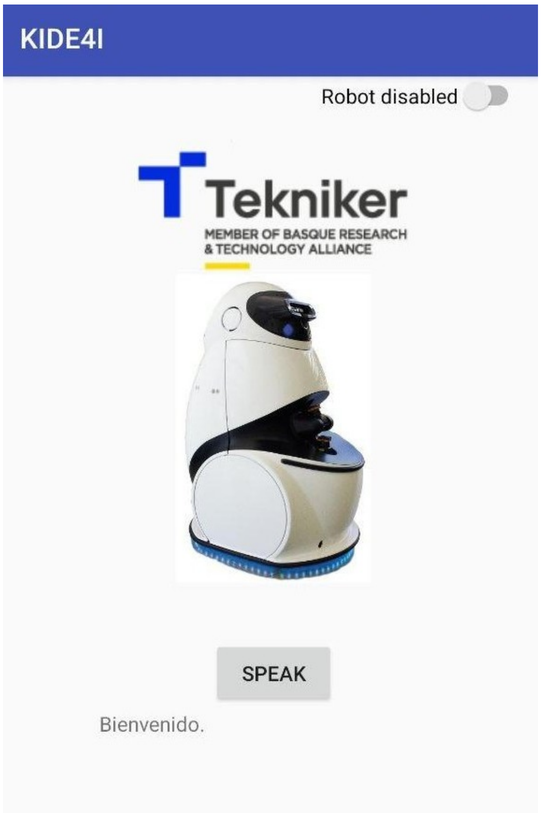
5. Experimental Setup

So as to evaluate and validate KIDE4I, and its adaptations KIDE4Guide and KIDE4BinPicking, two user studies have been defined and carried out.

To provide users with an interface to interact with the robot through voice commands, an Android application was developed, which is able to access the dialogue system(s) deployed in the server machine(s). The application consists of a button, `SPEAK`, used to interact with the robot, as it can be seen in Figure 5. For simplicity, users were provided with a mobile phone with the application installed, so it was not necessary for them to install it on their phones.

In each user study, 12 subjects—which were considered as potential users of these applications—were recruited. Each user was expected to perform at least 5 dialogues (a dialogue was considered a set of interactions in which the user conveys with the system a specific action to perform) according to a set of instructions given to them by the study personnel, where a short description of the scenario was provided, along with the type of interactions could be addressed to the system. After finishing their dialogues, each user was requested to fill a questionnaire. The questionnaire chosen to evaluate the dialogue system was the SASSI questionnaire [41], as it provides a comprehensive evaluation on speech-based dialogue systems and it is considered as an standard resource to evaluate such systems. However, for each use case, some additional questions were added to evaluate specific areas that are not covered by SASSI, such as security or productivity (*vid.* Appendix A). Users were reminded that the object of their evaluation was the dialogue system and not third-party components such as the app itself or the ASR technology or the robot, if present.

The following sections provide specific details for each of the user studies carried out in the context of this work.



Bienvenido.

Figure 5. Screenshot from the mobile application used to interact with KIDE4I.

5.1. Guide/Logistics Robot

In this study, users were expected to interact with the guide robot through voice commands in order to be guided to a destination of their choice. In this case, the presence of the robot was emulated (that is, the robot was not physically present). However, each time a dialogue was successfully completed, users received a simulation of their command being sent to the robot to reproduce the use case scenario in the most precise way possible.

The available destinations were defined in a set of maps that contained a selection of destinations that included laboratories, workshops, machines, people, spaces and other objects that corresponded to the Tekniker facility. These maps were provided to each of the users so as to perform their commands, along with some instructions about the experimentation itself, such as wording restrictions and the app's basic controls.

The commands directed to the dialogue system had no wording restrictions in general, which meant that destinations could be referred explicitly ("Take me to the vending machine") or implicitly ("I want to eat something"). However, some sequences were not supported (such as coordination of destinations—"I want to go to the meeting room **and** then to the toilet"), and users were instructed about them. The users did not receive any further instructions regarding the commands to direct towards the dialogue system so as to ensure a natural interaction and not interfere with their interactions.

Table 3 shows demographic data for the participants of the study. All subjects are familiar with technologies in general as they work in domains that require a knowledge of them.

Table 3. Demographic data for participants in the KIDE4Guide user study. (**a**) Gender information. (**b**) Age information. (**c**) Frequency of voice interaction with everyday devices.

(a)	Gender	(b)	Age	(c)	Interaction with Everyday Devices
M	33%	**24–34**	58%	**Never**	33.3%
F	42%	**35–44**	17%	**Sometimes**	58.3%
N/D	25%	**45–52**	17%	**Frequently**	8.3%
		N/D	8%		

5.2. Bin-Picking Robot

In this user study, the objective was to interact with the robot though voice commands to indicate which cartridges would go to one of the containers of choice, whereas the rest were to be placed into the other container.

In this case, the robot, depicted in Figure 4, was physically present in a manufacturing laboratory. Due to this, users were provided with an industrial headset with microphone, designed to prevent ambient noise to interfere with the voice captured and to protect them from said noise.

Before the experimentation, users were instructed about the objectives of the study and minimal interaction restrictions. As in the previous user study, there were not wording restrictions in general, except for coordination of targets (e.g., "Put the black ones into container 1 **and** the yellow ones into container 2"). Furthermore, they were shown which cartridges were available and how they looked like, so they would see if the robot was correctly performing the action it was ordered to execute.

During the experimentation, and for each interaction, users were asked to choose three or four cartridges, which were placed on the platform in front of the robot by the personnel in charge of the study. Then, they were asked to perform their commands using the app and the headset they were provided with from the position they were instructed to remain in. Figure 6 shows the conditions of the experimentation.

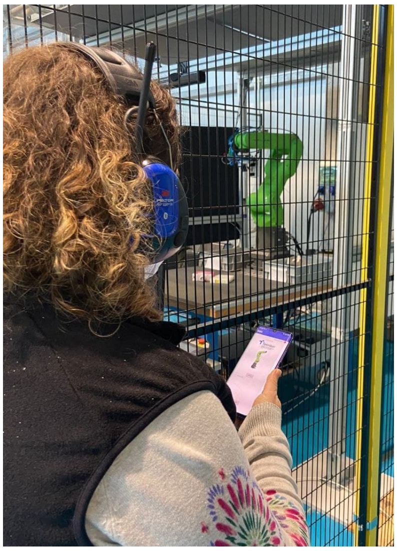


Figure 6. User interacting with the bin-picking robot according to the conditions defined for the experimentation.

Table 4 shows demographic information regarding the study participants. As in the previous user study, all participants were familiar with technologies in general and they are also related to some degree to industrial processes.

Table 4. Demographic data for participants in the KIDE4BinPicking user study. (**a**) Gender information. (**b**) Age information. (**c**) Frequency of voice interaction with everyday devices.

(a)	Gender	(b)	Age	(c)	Interaction with Everyday Devices
M	50%	**24–34**	42%	**Never**	8.3%
F	50%	**35–44**	33.3%	**Sometimes**	83.3%
		45–54	16.6%	**Frequently**	8.3%
		55–59	8%		

5.3. Results

So as to provide a comprehensive evaluation of the system through the user studies described previously, results will be reported at qualitative and quantitative level:

- Qualitative evaluation. At this level, responses from the SASSI questionnaire will be analyzed.
- Quantitative evaluation. Evaluation at this level will provide quantitative information on the systems evaluated by considering different units of analysis:
  - Dialogue. From this perspective, the dialogue as a whole (i.e., a series of interactions between the system and the user to achieve an executable action) is assessed. To do so, three aspects are evaluated:
    - Dialogue completion rate. Whether a dialogue has been successful or not [42].
    - Dialogue completion steps. How many steps were necessary to complete a dialogue.
    - Error analysis. Number of cases the user goal was not fulfilled by the system due to a specific reason in interpretation.
  - Interaction. Here, the interactions performed in each dialogue turn by the system and the user are evaluated considering the following information:
    - Response time. Time needed by the system to provide a response given a user request.

5.3.1. Qualitative Evaluation: SASSI Questionnaire

To provide a qualitative evaluation of the user studies reported in this work, the SASSI questionnaire was used, since it allows to comprehensively evaluate different aspects of KIDE4I in their different adaptations. Furthermore, so as to cover other areas that are relevant for this study regarding the system's industrial application, additional questions have been added to SASSI (*vid.* Appendix A). Thus, the aspects covered by the questionnaires are the following [41]:

- Response accuracy (SASSI). It refers to whether the system is able to interpret an user command and generates an appropriate response.
- Likeability (SASSI). It refers to the user perception of the system in terms of usefulness, pleasantness and friendliness.
- Cognitive demand (SASSI). It stands for the mental effort required by the user to interact with the system. In industrial scenarios, this aspect is especially relevant, since one of the main objectives of KIDE4I is to simplify the performance of specific tasks.
- Annoyance (SASSI). This aspect evaluates how repetitive or annoying is to interact with the system.
- Habitability (SASSI). It refers to whether the user knows what to say to the system.
- Speed (SASSI). It evaluates if the system response given a user interaction is fast.
- Verbosity. It determines whether the system interactions are too long. In this sense, the system should give the correct amount of information so as users can perform their tasks in the minimum amount of time.
- Productivity. This question aims to determine if the fact of using this system would increase the user's productivity, as it is also one of the main objectives of KIDE4I.

- Security. As user security is of utmost importance in industrial scenarios, this item evaluates if the design of the system allows users to perform the intended tasks by preserving a secure distance.

Each question in SASSI consists of a 6-point Likert scale, where 1 stands for *Strongly disagree* and 6 to *Strongly agree*.

Figures 7 and 8 provide the results obtained from the questionnaire for the SASSI questions and for the additional questions. In general, it can be observed that the results are very positive in both use cases, with median scores between 5 and 6 for common questions between use cases and 4 and 6 for additional questions for the bin-picking use case.

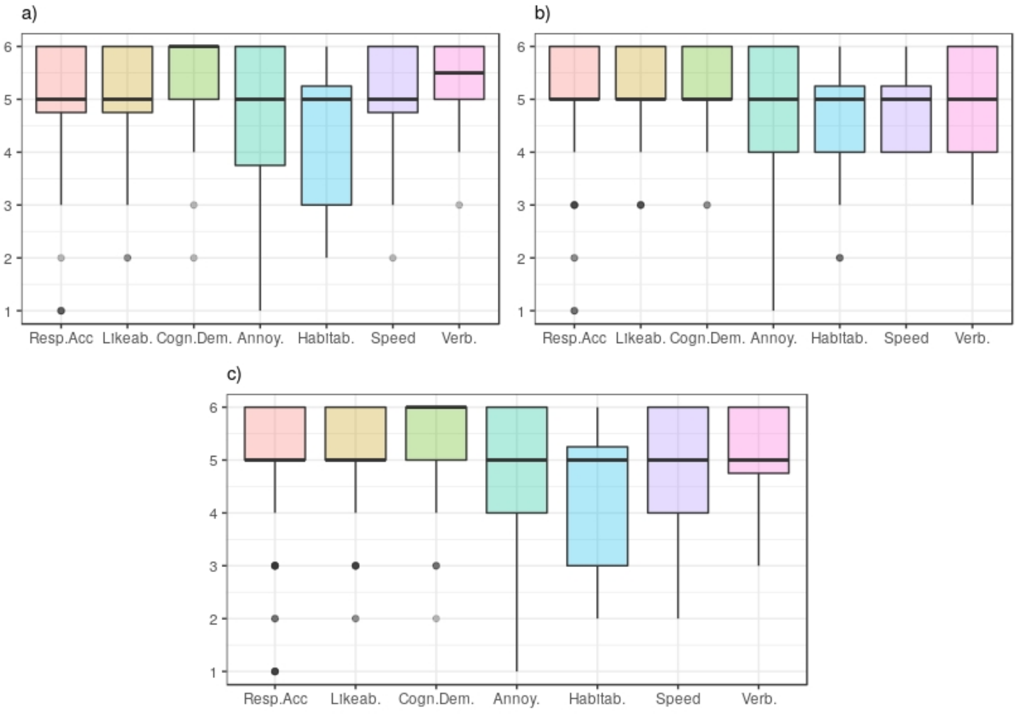


Figure 7. Results obtained from user questionnaires. (**a**) Results obtained for the guide use case. (**b**) Results obtained for the bin-picking use case. (**c**) Results obtained considering both experimentations.

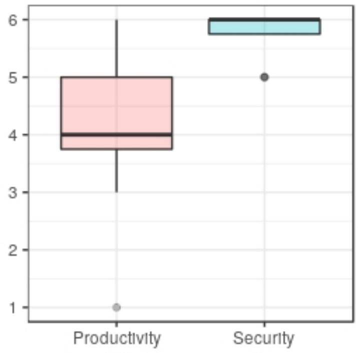


Figure 8. Results obtained from user questionnaires: additional questions for the bin-picking use case.

The results in Figure 7 show that some aspects had more consensus between participants than others. The aspects that denoted more variability were **annoyance**, **habitability** and **speed**. For **annoyance**, it can be observed that most answers are among the highest ratings, but also a significant amount of answers—in the sense that these observations are not outliers—have obtained the lowest ratings. This is caused by the results obtained for question number 24 ("*The interaction with the system is repetitive*"), which was rated with an average score of 2.38. Since the tasks to perform in the user studies were strongly related to industrial scenarios, which inherently consist of very specific actions on specific elements, this was an expected outcome. Regarding **habitability**, results obtained more average scores than the rest of the evaluation items. In this case, although they were given instructions about the task and interactions, participants were not sure about what to say to the system not because it was not clear, but because they were afraid the system would fail if they were *too natural* in their interactions. Finally, scores for **speed** were different between use cases. In the guide use case, results for this aspect were more variate, and for the bin-picking use case were more constant. Since the complexity of the guide use case was higher than in the bin-picking use case, in some cases KIDE4I needed more time to process user commands (*vid.* Section 5.3.3). However, users considered the system to be fast in general terms.

On the other hand, the **most appreciated** aspects among users were **response accuracy**, **likeability**, **cognitive demand** and **verbosity**, with more consensus between participants. The most relevant results are the ones obtained for **cognitive demand**, as this aspect refers to one of the main objectives of KIDE4I and validates the easiness of use of the system, which makes it highly suitable in industrial scenarios.

All in all, the system has been considered to be **accurate**, **useful**, **efficient**, **not demanding**, **flexible**, **fast** and that, in general, **the amount of information provided is correct**.

For additional questions for the bin-picking use case—in Figure 8—they have also been evaluated very positively, and it is specially relevant that the feeling of security by using this system is very high, reaching nearly a perfect score among users. Furthermore, these results show that most users consider that the system would be a plus in their productivity.

Table 5 shows the scores for the different evaluated aspects in the two user studies, averaged over the three age groups stated in Tables 3 and 4 (24–34, 35–44, 45–54). The table shows that the scores given by participants of different ages is roughly the same, with relatively small standard deviation values. However, and despite the small standard deviation values, these figures show some tendencies according to age. In general, the 24–34 age group assessed more positively the evaluated aspects, except for verbosity and annoyance, in which the results show the opposite, being the 45–54 group the most satisfied. This can be associated to the fact that older people tend to appreciate to be provided a good amount of information, whereas younger people prefer a fast execution rather than information.

Table 5. Scores and standard deviation values over three different age groups (24–34, 35–44, 45–54) for the two user studies reported in this paper.

Aspect	24–34	35–44	45–54	Total Average	Standard Deviation
Response Accuracy	5.07	4.52	4.68	4.75	0.29
Likeability	5.21	4.94	4.95	5.03	0.15
Cognitive Demand	5.45	5.30	5.43	5.40	0.08
Annoyance	4.59	4.31	4.80	4.57	0.25
Habitability	4.62	4.38	4.35	4.45	0.15
Speed	5.17	4.55	4.63	4.78	0.34
Verbosity	4.75	5.15	5.42	5.11	0.34

5.3.2. Quantitative Evaluation at Dialogue Level

This level of analysis aims to evaluate the dialogue system adaptations involved in the user studies by whether the **interaction goal has been fulfilled** (**dialogue completion**), **how many turns** did it take for that interaction goal to be reached (**dialogue steps**) and the number and classification of the errors that caused a dialogue not to be successful or to require some reformulation from the user (**error analysis**).

To assess **dialogue completion**, all dialogues have been analysed by a group of experts, who were expected to determine whether the user goal was successfully **completed** or **not**

completed. In the case of completed tasks, dialogues were classified between *fully completed* or *partially completed*, depending on whether the user had to reformulate the query at some point of the dialogue (see Example (8)).

(8) a. **Initial—not correctly interpreted:** "Tengo sed"
"I am thirsty"
b. **Reformulation—correctly interpreted:** "Quiero beber"
"I want to drink"

Table 6 shows the percentage of the dialogue completion rates for each use case. As it can be observed, the results are very positive, where the successful completion rates reach an 84.34% and 82.67%, respectively. In the guide use case, an 8.43% of these dialogues were classified as partially completed.

Table 6. Dialogue completion results for the user studies reported in this work. Values in parentheses stand for dialogues classified as *partially completed*.

	KIDE4Guide		**KIDE4BinPicking**	
	%	#	%	#
Completed	84.34 (8.43)	70 (7)	82.67	62
Not completed	15.66	13	17.33	13
Total		83		75

Regarding **dialogue steps**, Table 7 shows the average number of steps required to complete the dialogue. It is important to keep in mind that each dialogue includes 1 or 2 steps that are included by default in each adaptation as part of their design: in the KIDE4Guide case, the system initiates the dialogue by presenting itself and, in both KIDE4Guide and KIDE4BinPicking, when an action is obtained, the system asks the user for confirmation. Come as it may, the number of dialogues required to achieve the user's goal is positive enough to determine that KIDE4I allows an agile interaction between the user and the system.

Table 7. Number of average number of steps to successfully complete a dialogue. Values in parentheses stand for values that exclude steps implemented by default.

KIDE4Guide	**KIDE4BinPicking**
5.4 (3.4)	4.3 (3.3)

Finally, as a necessary step to improve the system in future versions, the dialogues that did not fulfill the goal of the user or required reformulations have been analysed to identify the **source of the errors** that led to unsuccessful interpretations. The errors identified are the following:

- Automatic Speech Recognition (ASR). Not accurate transcriptions (Example (9)).

  (9) a. **Obtained:** "Quiero *unas alas* con algún ordenador"
  "I want *a pair of wings* with some computer"
  b. **Correct:** "Quiero una sala con algún ordenador"
  "I want a room with some computer"

- Syntactic analysis. Structures that are not correctly analyzed—Example (10)—or words with wrong lemmas (usually for words that are not in the tool's dictionary)—Example (11).

  (10) a. **Obtained:** "El contenedor 2 es $para_{verb}$ la marca Canon"
  "Container 2 is stop the Canon brand"

b. **Correct:** "El contenedor 2 es *para*$_{preposition}$ la marca Canon"
"Container 2 is for the Canon brand"

(11) a. **Obtained:** "Pon los cartuchos HP$_{lemma:``h\_p''}$ en el contenedor 2"
"Put the HP cartridges in container 2"

b. **Correct:** "Pon los cartuchos HP$_{lemma:``hp''}$ en el contenedor 2"
"Put the HP cartridges in container 2"

- Rules. Structures that have not been considered in the definitions and/or rules.
- Polarity interpreter. Classification errors in the polarity interpreter component (Example (12)).

(12) a. **Obtained:**
SYSTEM: "¿Quieres que te guíe hacia la cafetera?"
"Do you want me to guide you to the coffee machine?"
USER: "Efectivamente$_{polarity:NO}$"
"Indeed"

b. **Correct:**
SYSTEM: "¿Quieres que te guíe hacia la cafetera?"
"Do you want me to guide you to the coffee machine?"
USER: "Efectivamente$_{polarity:YES}$"
"Indeed"

- Ontology-related errors. Errors in both the ontology modelling or the way information is retrieved from the ontology.

Table 8 shows the number of cases for the identified error sources that lead to not completed dialogues in both user studies. As it can be seen, in KIDE4Guide the typology of errors is more varied than in KIDE4BinPicking, being the most common errors the ones related with the modelling of the ontology. This is due to the higher complexity of the KIDE4Guide scenario, since this adaptation includes a wide variety of spaces, the elements contained in them, and the fact that it is possible to refer implicitly to spaces (e.g., "I want to *eat*" → *vending machine*). In KIDE4BinPicking, however, errors predominantly stemmed from incorrect syntactic analyses of user commands. More specifically, it had to do with one of the brands involved, HP, the lemma of which was obtained incorrectly due to the fact that it was not included in the tool's dictionary.

Table 8. Sources of the errors observed in the *partially completed* and *not completed* dialogues performed in the user studies.

	KIDE4Guide		**KIDE4BinPicking**	
	%	#	%	#
ASR	20	4	-	-
Syntactic analysis	10	2	92	12
Rules	-	-	8	1
Polarity interpreter	5	1	-	-
Ontology-related	65	13	-	-
Total		20		13

After observing and analysing these errors, the rules and ontology-related ones have been solved. For syntactic analysis, required adaptations have been performed in Freeling's dictionaries. Regarding ASR and the polarity interpreter, although the impact of these errors is not critical, new tools are expected to be explored as part of future work.

5.3.3. Quantitative Evaluation at Interaction Level: Response Time

Table 9 includes the average time of response of both KIDE4Guide and KIDE4BinPicking. As it can be observed, the response times for each adaptation are very fast, which are endorsed by the results obtained from the user questionnaires for the speed aspect, reported in Section 5.3.1. However, it is worth discussing the difference in response times between use cases. On the one hand, the guide use case is more complex in its modelling, and therefore a single destination can be referred in different ways, explicitly or implicitly (for example, through an object contained in said destination or an action that can be performed in it). Due to this variety of interactions, the response times may increase. This can be seen in Examples (13) and (14), where a television is required, but it is referred differently: Example (13) sets a restriction on all the meeting rooms available, whereas (14) just requests for a television. Between the 2 examples there is a difference of 0.3 s, which is explained by the difference in complexity between requests.

Table 9. Average time of response for each use case.

KIDE4Guide	KIDE4BinPicking
1.25 s	0.75 s

These examples also show that confirmations/negations also require a very low amount of time to process.

(13) a. **User:** “Quiero una sala con televisión”
"I want a room with a TV"

b. **System:** “Tengo demasiadas opciones que se corresponden con *sala con televisión*. ¿Te parece bien que te guíe hacia la puerta de la sala de reunión Sua 2?”
"I have too many options that correspond with *room with a tv*. Is it okay if I take you to the entrance of meeting room Sua 2?” [**Time elapsed:** 2.17 s]

c. **User:** “sí”
“yes”

d. **System:** “¡Ya está! ¿Necesitas algo más?”
“Done! Do you need anything else?” [**Time elapsed:** 0.19 s]

(14) a. **User:** “Quiero ver la televisión”
"I want to watch TV”

b. **System:** “Tengo demasiadas opciones que se corresponden con *televisión*. ¿Te parece bien que te guíe hacia la puerta de la sala de reunión Aura 1?”
"I have too many options that correspond with *TV*. Is it okay if I take you to the entrance of meeting room Aura 1?” [**Time elapsed:** 1.87 s]

c. **User:** “sí”
“yes”

d. **System:** “¡Ya está! ¿Necesitas algo más?”
“Done! Do you need anything else?” [**Time elapsed:** 0.16 s]

As for the bin-picking use case, the complexity of the scenario is simpler and, thus, the time required to process user commands is lower. This can be observed in Examples (15) and (16). In Example (15), the system only takes 0.87 s to process the initial command and to detect that there is one argument (the destination container) missing and, in Example (16), if the user command contains all the necessary information, it takes 1.11 s to process the whole command. As in the guide use case, affirmations/negations take around 0.2 s to process.

(15) a. **User:** "Coge el cartucho Brother"
"Pick the Brother cartridge"

b. **System:** "Hay información que me falta. Necesito que me digas el contenedor donde quieres depositar el cartucho, por favor."
"I am missing some information. Tell me the container you want to place the cartridge, please." [**Time elapsed:** 0.87 s]

c. **User:** "En el contenedor 2"
"In container 2"

d. **System:** "Entonces, quieres que coja y ponga los cartuchos del tipo Brother en 2, ¿verdad?"
"Then, you want me to pick and place the Brother cartridges in 2, isn't it?" [**Time elapsed:** 0.21 s]

(16) a. **User:** "Lleva el magenta al 2"
"Take the magenta one to 2"

b. **System:** "Entonces, quieres que coja y ponga los cartuchos del tipo magenta en 2, ¿verdad?"
"Then, you want me to pick and place the magenta cartridges in 2, isn't it?" [**Time elapsed:** 1.11 s]

c. **User:** "sí"
"yes"

d. **System:** [Command sent to robot] [**Time elapsed:** 0.21 s]

These examples also show that the initial command is the command that takes longer to process, as the system also needs to obtain the action to perform and it usually contains most—if not all—information to process.

6. Discussion and Future Work

With the aim of assuring human workers' wellbeing when performing their everyday tasks, this work presents KIDE4I (Knowledge-drIven Dialogue framEwork for Industry), a task-oriented dialogue system framework for industrial scenarios, with ontologies—for domain modelling and dialogue management—as its core. The main characteristics of this system are that (i) it allows a natural communication between workers and industrial assets, reducing the cognitive demand to do so, (ii) it does not need large amounts of training data to be constructed, and (iii) its architecture is generic enough to adapt it to new use cases with a reduced amount of effort.

This paper also reports the methodology to adapt KIDE4I for its use with different applications and, to validate it, the adaptation process for 4 different use cases, all relevant in industrial contexts, has been described. This adaptation process shows that other adaptations benefit from the developments carried out for a base use case, especially in the ontology modelling and instantiation phase, in which more than a 90% of the classes and more than a 70% of the instances needed can be reused or obtained automatically.

To evaluate and validate KIDE4I, two user studies have been carried out, one for the KIDE4Guide adaptation (for a guide/logistics robot) and another for the KIDE4BinPicking adaptation (for a bin-picking robot). The results of these user studies have been reported in terms of qualitative and quantitative evaluation. Qualitative evaluation has been carried out through SASSI, a standardized questionnaire to evaluate spoken dialogue systems. The results on SASSI report very high scores in general (between 5 and 6 out of 6) for all the evaluation aspects considered and have revealed that the system is accurate, fast, useful, efficient, and, most importantly, not demanding cognitively and that users consider that using it would improve their productivity and their security.

As for quantitative evaluation, in more than an 80% of the dialogues in both use cases, the interaction goal was fulfilled, and the number of steps necessary to do so is around 3. Also, the dialogues that did not fulfil the interaction goal were analysed and classified according to the type of errors that caused these dialogues to be unsuccessful. For the KIDE4Guide adaptation, most errors were related to ontology modelling, and for KIDE4BinPicking, to the syntactic analysis performed in the key element extraction component. Once these errors were detected, they were solved. For the errors depending on third-party tools, such as the ASR technology or the libraries used for the polarity component, other options will be explored as future work in order to reduce the number of errors caused by these modules. Finally, the average time of response has been obtained for both use cases, and it has been observed that the more complex the use case, the more average time it takes to obtain an interpretation, although the differences between complex use cases (KIDE4Guide) and simpler ones (KIDE4BinPicking) is practically imperceptible to users (half a second of difference in average).

Furthermore, and to wrap up, the quantitative results obtained totally endorse the participants' experience reflected in the questionnaires.

Future work includes user studies for the remaining two KIDE4I adaptations and the improvements on functionalities provided by third-party tools mentioned above. In the long run, so as the system is able to learn from new interactions, a component that obtains feedback from users will be developed and implemented.

Author Contributions: Methodology, C.A., I.F., A.S.; software, C.A.; validation, C.A.; formal analysis, C.A., I.F., A.S.; investigation, C.A.; resources, C.A.; writing—original draft preparation, C.A.; writing—review and editing, C.A., I.F., A.S.; supervision, I.F., A.S.; experimental design, C.A., I.F., A.S.; experimentation supervision, C.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Basque Government's Elkartek research and innovation program, projects EKIN (grant no KK-2020/00055) and DeepText (grant no KK-2020/00088).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data can be consulted upon request to authors.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. The Subjective Assessment of Speech System Interfaces Questionnaire (SASSI)

This appendix shows the SASSI questionnaire given to the subjects of both user studies. Each question consists of a 6-point Likert scale, where 1 stands for *Strongly disagree* and 6 to *Strongly agree*.

As described in Section 5, extra questions were added in each user study to evaluate some areas that were not covered by SASSI. Some of these questions were extracted from other questionnaires (*Verbosity* and *Productivity*, from the SUISQ [43] questionnaire) and others were manually created (*Security*). So as to be integrated into the original SASSI questions, these extra questions were also evaluated by using a 6-point Likert scale.

It is important to note that the statements in SASSI can have positive ("The system is accurate") or negative ("I felt tense using the system") connotations. So as to obtain consistent evaluations, it is necessary to rescale negative statements so as to be considered as positive. For example, if a negative statement has a score of 1, its rescaled score would be 6 [44].

For understandability reasons, the questions in this appendix are in English. However, a Spanish version, translated by an expert, was provided to the study subjects.

Response Accuracy

1. The system is accurate.
2. The system is unreliable.
3. The interaction with the system is unpredictable.
4. The system didn't always do what I wanted.
5. The system didn't always do what I expected.
6. The system is dependable.
7. The system makes few errors.
8. The interaction with the system is consistent.
9. The interaction with the system is efficient.

Likeability

10. The system is useful.
11. The system is pleasant.
12. The system is friendly.
13. I was able to recover easily from errors.
14. I enjoyed using the system.
15. It is clear how to speak to the system.
16. It is easy to learn to use the system.
17. I would use this system.
18. I felt in control of the interaction with the system.

Cognitive Demand

19. I felt confident using the system.
20. I felt tense using the system.
21. I felt calm using the system.
22. A high level of concentration is required when using the system.
23. The system is easy to use.

Annoyance

24. The interaction with the system is repetitive.
25. The interaction with the system is boring.
26. The interaction with the system is irritating.
27. The interaction with the system is frustrating.
28. The system is too inflexible.

Habitability

29. I sometimes wondered if I was using the right word.
30. I always knew what to say to the system.
31. I was not always sure what the system was doing.
32. It is easy to lose track of where you are in an interaction with the system.

Speed

33. The interaction with the system is fast.
34. The system responds too slowly.

Extra: Verbosity—both UCs

35. I felt like I had to wait too long for the system to stop talking so I could respond.

Extra: Productivity—bin-picking UC

36. The system would help me be productive.

Extra: Security—bin-picking UC

37. This system allows me to interact with the robot from a secure distance without problems.

References

1. Madonna, M.; Monica, L.; Anastasi, S.; Di Nardo, M. Evolution of Cognitive Demand in the Human-Machine Interaction Integrated with Industry 4.0 Technologies. *WIT Trans. Built Environ.* **2019**, *189*, 13–19.
2. Kildal, J.; Fernández, I.; Lluvia, I.; Lázaro, I.; Aceta, C.; Vidal, N.; Susperregi, L. Evaluating the UX Obtained from a Service Robot that Provides Ancillary Way-Finding Support in an Industrial Environment. In *Advances in Manufacturing Technology XXXIII, Proceedings of the 17th International Conference on Manufacturing Research, Belfast, UK, 10–12 September 2019*; IOS Press: Amsterdam, The Netherlands, 2019; Volume 9, p. 61.
3. Budzianowski, P.; Wen, T.H.; Tseng, B.H.; Casanueva, I.; Ultes, S.; Ramadan, O.; Gašić, M. MultiWOZ—A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. *arXiv* **2018**, arXiv:1810.00278.
4. Luckow, A.; Cook, M.; Ashcraft, N.; Weill, E.; Djerekarov, E.; Vorster, B. Deep Learning in the Automotive Industry: Applications and Tools. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), San Francisco, CA, USA, 27 June–2 July 2016; pp. 3759–3768. [CrossRef]
5. Jurafsky, D.; Martin, J.H. Speech and Language Processing (Draft). 2021; Chapter 24. Available online: https://web.stanford.edu/~jurafsky/slp3/ (accessed on 20 January 2022).
6. Bugmann, G.; Pires, J.N. Robot-by-voice: Experiments on Commanding an Industrial Robot Using the Human Voice. *Ind. Robot. Int. J.* **2005**, *32*, 505–511.
7. Veiga, G.; Pires, J.; Nilsson, K. Experiments with Service-Oriented Architectures for Industrial Robotic Cells Programming. *Robot. Comput.-Integr. Manuf.* **2009**, *25*, 746–755. [CrossRef]
8. Aceta, C.; Fernández, I.; Soroa, A. TODO: A Core Ontology for Task-Oriented Dialogue Systems in Industry 4.0. In *Further with Knowledge Graphs*; IOS Press: Amsterdam, The Netherlands, 2021; pp. 1–15.
9. Ward, W.; Issar, S. *Recent Improvements in the CMU Spoken Language Understanding System*; Technical Report; School of Computer Science, Carnegie-Mellon University: Pittsburgh, PA, USA, 1994.
10. Wei, Z.; Liu, Q.; Peng, B.; Tou, H.; Chen, T.; Huang, X.J.; Wong, K.F.; Dai, X. Task-Oriented Dialogue System for Automatic Diagnosis. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 2, pp. 201–207.
11. Goddeau, D.; Meng, H.; Polifroni, J.; Seneff, S.; Busayapongchai, S. A Form-Based Dialogue Manager for Spoken Language Applications. In Proceedings of the Fourth International Conference on Spoken Language Processing (ICSLP '96), Philadelphia, PA, USA, 3–6 October 1996; Volume 2, pp. 701–704.
12. Lee, S.; Eskenazi, M. Recipe For Building Robust Spoken Dialog State Trackers: Dialog State Tracking Challenge System Description. In Proceedings of the SIGDIAL 2013 Conference, Metz, France, 23–24 August 2013; pp. 414–422.
13. Lee, S. Structured Discriminative Model for Dialog State Tracking. In Proceedings of the SIGDIAL 2013 Conference, Metz, France, 23–24 August 2013; pp. 442–451.
14. Williams, J.D. Multi-Domain Learning and Generalization in Dialog State Tracking. In Proceedings of the SIGDIAL 2013 Conference, Metz, France, 23–24 August 2013; pp. 433–441.
15. Mrkšić, N.; Séaghdha, D.O.; Thomson, B.; Gašić, M.; Su, P.H.; Vandyke, D.; Wen, T.H.; Young, S. Multi-Domain Dialog State Tracking Using Recurrent Neural Networks. *arXiv* **2015**, arXiv:1506.07190.
16. Henderson, M.; Thomson, B.; Young, S. Deep Neural Network Approach for the Dialog State Tracking Challenge. In Proceedings of the SIGDIAL 2013 Conference, Metz, France, 23–24 August 2013; pp. 467–471.
17. Chen, H.; Liu, X.; Yin, D.; Tang, J. A Survey on Dialogue Systems: Recent Advances and New Frontiers. *SIGKDD Explor. Newsl.* **2017**, *19*, 25–35. [CrossRef]
18. Aceta, C.; Kildal, J.; Fernández, I.; Soroa, A. Towards an Optimal Design of Natural Human Interaction Mechanisms for a Service Robot with Ancillary Way-Finding Capabilities in Industrial Environments. *Prod. Manuf. Res.* **2021**, *9*, 1–32. [CrossRef]
19. Suendermann, D.; Evanini, K.; Liscombe, J.; Hunter, P.; Dayanidhi, K.; Pieraccini, R. From Rule-Based to Statistical Grammars: Continuous Improvement of Large-Scale Spoken Dialog Systems. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 4713–4716.
20. Gustavsson, P.; Syberfeldt, A.; Brewster, R.; Wang, L. Human-Robot Collaboration Demonstrator Combining Speech Recognition and Haptic Control. *Procedia CIRP* **2017**, *63*, 396–401. [CrossRef]
21. Stenmark, M.; Nugues, P. Natural Language Programming of Industrial Robots. In Proceedings of the IEEE ISR 2013, Seoul, Korea, 24–26 October 2013; pp. 1–5.

22. Maurtua, I.; Fernández, I.; Tellaeche, A.; Kildal, J.; Susperregi, L.; Ibarguren, A.; Sierra, B. Natural Multimodal Communication for Human–Robot Collaboration. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1–12. [CrossRef]
23. Kingsbury, P.R.; Palmer, M. From TreeBank to PropBank. In Proceedings of the LREC, Las Palmas de Gran Canaria, Spain, 29–31 May 2002; European Language Resources Association (ELRA): Paris, France, 2002; pp. 1989–1993.
24. Antonelli, D.; Bruno, G. Human-Robot Collaboration Using Industrial Robots. In Proceedings of the 2nd International Conference on Electrical, Automation and Mechanical Engineering, Shenzhen, China, 17–18 September 2017; Atlantis Press: Paris, France, 2017; pp. 99–102.
25. Yakoub, M.S.; Selouani, S.A.; Nkambou, R. Mobile Spoken Dialogue System Using Parser Dependencies and Ontology. *Int. J. Speech Technol.* **2015**, *18*, 449–457. [CrossRef]
26. Altinok, D. An Ontology-Based Dialogue Management System for Banking and Finance Dialogue Systems. *arXiv* **2018**, arXiv:1804.04838.
27. Wessel, M.; Acharya, G.; Carpenter, J.; Yin, M. OntoVPA-an Ontology-Based Dialogue Management System for Virtual Personal Assistants. In *Advanced Social Interaction with Agents*; Springer: New York, NY, USA, 2019; pp. 219–233.
28. Teixeira, M.S.; Maran, V.; Dragoni, M. The Interplay of a Conversational Ontology and AI Planning for Health Dialogue Management. In Proceedings of the 36th Annual ACM Symposium on Applied Computing, Gyeongju, Korea, 22–26 March 2021; pp. 611–619.
29. OpenLink Software. Virtuoso [Software]. Available online: https://virtuoso.openlinksw.com/ (accessed on 20 January 2022).
30. Carreras, X.; Chao, I.; Padró, L.; Padró, M. FreeLing: An Open-Source Suite of Language Analyzers. In Proceedings of the LREC, Lisbon, Portugal, 26–28 May 2004; pp. 239–242.
31. Hulden, M. Foma: A Finite-State Compiler and Library. In Proceedings of the Demonstrations Session at EACL 2009, Athens, Greece, 3 April 2009; pp. 29–32.
32. Hofman, E. *senti-py*: A Pre-Trained Sentiment Analysis Classifier in Spanish. Available online: https://github.com/aylliote/senti-py (accessed on 20 January 2022).
33. Keet, M. An Introduction to Ontology Engineering. 2020; Volume 1. Available online: https://people.cs.uct.ac.za/~mkeet/files/OEbook.pdf (accessed on 20 January 2022).
34. Aceta, C.; Fernández, I.; Soroa, A. Ontology Population Reusing Resources for Dialogue Intent Detection: Generic and Multilingual Approach. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), Online, 1–3 September 2021; pp. 10–18.
35. Romero, D.; Stahre, J.; Wuest, T.; Noran, O.; Bernus, P.; Fast-Berglund, A.; Gorecky, D. Towards an Operator 4.0 Typology: A Human-Centric Perspective on the Fourth Industrial Revolution Technologies. In Proceedings of the International Conference on Computers and Industrial Engineering (CIE46), Tianjin, China, 29–31 October 2016; pp. 29–31.
36. Brickley, D. Basic Geo (WGS84 lat/long) Vocabulary. Version 1.21. Available online: http://www.w3.org/2003/01/geo/wgs84_pos# (accessed on 20 January 2022).
37. Brickley, D.; Miller, L. FOAF Vocabulary. Version 0.99. Available online: http://xmlns.com/foaf/spec/ (accessed on 20 January 2022).
38. Rodriguez-Castro, B.; Torok, L.; Hepp, M. Printer Vocabulary Ontology. Available online: http://purl.org/opdm/printer# (accessed on 20 January 2022).
39. del Pozo, A.; García-Sardiña, L.; Serras, M.; González-Docasal, A.; Torres, M.I.; Ruiz, E.; Fernández, I.; Aceta, C.; Konde, E.; Aguinaga, D.; et al. EKIN: Towards Natural Language Interaction with Industrial Production Machines. In *Annual Conference of the Spanish Association for Natural Language Processing 2021: Projects and Demonstrations*; CEUR: Málaga, Spain, 2021; pp. 5–8.
40. Fernández, I.; Casla, P.; Esnaola, I.; Parigot, L.; Marguglio, A. Towards Adaptive, Interactive, Assistive and Collaborative Assembly Workplaces through Semantic Technologies. Preprint. 2020. Available online: https://www.researchgate.net/publication/344362531_Towards_Adaptive_Interactive_Assistive_and_Collaborative_Assembly_Workplaces_through_Semantic_Technologies (accessed on 20 January 2022).
41. Hone, K.S.; Graham, R. Towards a Tool for the Subjective Assessment of Speech System Interfaces (SASSI). *Nat. Lang. Eng.* **2000**, *6*, 287–303. [CrossRef]
42. Wu, W.; Guo, Z.; Zhou, X.; Wu, H.; Zhang, X.; Lian, R.; Wang, H. Proactive Human-Machine Conversation with Explicit Conversation Goals. *arXiv* **2019**, arXiv:1906.05572.
43. Polkosky, M.D. Toward a Social-Cognitive Psychology of Speech Technology: Affective Responses to Speech-Based e-Service. Ph.D. Thesis, University of South Florida, Tampa, FL, USA, 2005.
44. Olaso Fernández, J.M. Spoken Dialogue Systems: Architectures and Applications. Ph.D. Thesis, Euskal Herriko Unibertsitatea, Leioa, Spain, 2017.