Implementing Diverse Robotic Interactive Systems Using VOnDA

Bernd Kiefer[1] , **Christian Willms**[2]
[1]paragon semvox GmbH [2]Deutsches Forschungsinstitut für Künstliche Intelligenz (DFKI GmbH)
kiefer@semvox.de, christian.willms@dfki.de

Abstract

This paper describes the implementation of different interactive robotic systems using the dialogue framework VOnDA, an interactive robotic arm, a guidance and a transport robot developed in the INTUITIV project, and the social robotic companion and tutor of the PAL system. The applications differ in terms of complexity in multiple dimensions, e.g., concerning dialogue structure, frequency of sensory input, architecture (embedded vs. cloud), etc. We highlight the different requirements, and aim to demonstrate the flexibility of the framework and its feasibility in multiple contexts and on different levels of control, from mere interaction modelling to being the central agent control component.

1 Introduction

Interactive robotic applications come in many flavours, from speech or multi-modal command and control interfaces for industrial robots [SynDiQuAss Team, 2020; Mocan *et al.*, 2016; Ramgire and Jagdale, 2016] to social companion robots [ASAROB Team, 2020; Foster *et al.*, 2016] or interactive learning companions for different areas [Belpaeme *et al.*, 2013; PAL, 2018; Belpaeme *et al.*, 2015]. The different applications have quite diverse requirements in terms of reactivity, sensor input and processing, interaction capabilities, user adaptivity or long-term interaction abilities.

In this paper, we present different applications of interactive robotic agents implemented with the VOnDA framework[Kiefer *et al.*, 2019], which was initially designed for dialogue management, but turned out to be able to serve as a basic management layer for many kinds of interactive robotic (and other) systems.

1.1 Related Work

In this section, for reasons of space, we will concentrate on interactive robotic applications similar to the ones we are going to describe later. There are many competitive frameworks available, e.g. OpenDial [Lison and Kennington, 2016], which takes a similar hybrid approach (more are discussed in [Kiefer *et al.*, 2019]), with less focus on long-term memory. Some are already embedded into a robot ecosystem, such as *rrSDS* [Kennington *et al.*, 2020], a feature that VOnDA is currently lacking.

ASARob The aim of the research project "Aufmerksamkeitssensitiver Assistenzroboter" (ASARob)[1], performed by Fraunhofer IAB, is to develop new skills for interactive assistance robots that allow to detect attention state of their counterpart and, if necessary, influence and direct it through their behaviour. Using different robot sensors and computing location and motions of the user, including gaze, head direction and posture, it recognises where the user's attention is directed and what his/her possible intention might be. Verbal interactions are another source of information and play an important role in regaining the user's attention on shared journeys. In addition, stylised facial expressions and movements of the robot are used for this purpose. The multimodal interaction component in ASARob is implemented by the geni:OS[2] platform of paragon semvox, which shares aspects such as ontological modelling of interactions and reactive rules with VOnDA, but is commercially available only.

MuMMER The goal of this project [Foster *et al.*, 2019] is to build a **Mu**ltimodal **M**all **E**ntertainment **R**obot, that provides help and guidance to customers, as well as marketing activities. The challenge is that the robot has to be socially appropriate and entertaining to gain broader acceptance. It uses verbal (speech-based) and non-verbal communication fused with appropriate motions to achieve this. Multiple sensors track people in the vicinity, their location, gaze and social signals and share their results with the interaction modules to generate modulated motions for expressing non-verbal social signals. The conversational system is built upon several machine learning based modules which are nevertheless based on dialogue acts and frame semantics to achieve functionalities for social talk that go beyond usual slot-filling dialogues employed in voice agents. To synchronise verbal utterances and motions, which may be quite different in duration, their execution system uses *recipes* that tie dialogue and physical actions together, and the execution system has been extended to manage multi-threaded parallel executions and their arbitration, including the possibility that users interrupt such a process and switch to a completely different action. The robot is fully autonomous, albeit using heavy computing machinery

[1]https://asarob.de/
[2]https://www.semvox.de/technologien/genios-3/

to accomplish the different complex computations for sensory and other modules, and was tested in a three-month user study. The dialogue and agent control system of this project is very different from a VOnDA implemented agent and therefore hard to compare, but the agents described in sections 3.2 and 3.3 share a lot of the demanding aspects of this system.

2 Background: The VOnDA framework

VOnDA uses a hybrid rule-based and statistical approach to be able to trade control against flexibility depending on the demands of the application. This covers the range from fully deterministic systems to socially interactive robots, where central requirements are extensive user adaptivity and variation in then interaction, including some surprise or unpredictability.

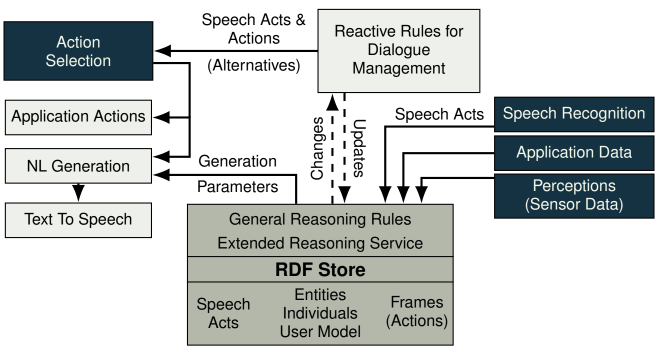

Figure 1: Architecture of a VOnDA Agent

In figure 1, which shows the schematic of a VOnDA agent, the statistical component is in the box labelled *action selection*. It can be implemented by probabilistic models of any complexity, from Bayes Nets to deep neural network solutions, and has access to all the data available in the information state, if needed.

The information state is kept in an extended *RDF Store* and reasoner (HFC, [Krieger, 2013; Krieger, 2014]) which allows to efficiently attach time information for all facts in the data base and to reason about them. In this way, the information state contains a full history of past interactions, even beyond a single session, and all relevant system and contextual data for dealing with a user in a socially appropriate and pleasant form. Data can be analysed and aggregated over this history to implement interaction patterns that foster long-term use of such a system, e.g. by showing recollection or familiarity with the current user. On the other hand, the semantic long term history also allows to model gradual forgetting on any scale, e.g., by specifying reasoning rules that influence the validity of data chunks depending on their class and lifetime. For dealing efficiently with, e.g., high-frequency sensor data, the RDF store has support for *streaming reasoning*, which can also be used for aggregating modules like an *episodic memory*.

The use of an RDF store as a central database has other advantages: Firstly, the specification of application data structures, such as sensor data, natural language semantics or other internal data *and* the data objects itself are represented in the same format. We will describe in the next section how VOnDA uses for example the type and property system of OWL to support a compact rule specification language, and the time information to obtain modifiable object in the database. Secondly, reasoning rules and queries can be used to perform complex tasks in a more declarative way. HFC allows to add custom reasoning rules, which gives a lot of flexibility, but has to be handled with care since it may introduce performance non-termination problems.

Relevant changes in the information state, which are determined by efficient filters similar to those used for streaming reasoning, trigger the rule system of a VOnDA agent. In combination with the efficient processing of sensory and other input data, highly reactive systems with very low response times are possible.

The VOnDA framework uses shallow semantic structures commonly used in dialogue systems for natural language understanding (NLU) as well as generation (NLG). The advantage of this symbolic representation for generation is that the output (verbal and non-verbal, e.g., motions) can be flexibly modulated for different situations, depending on information state data, such as the emotional state of the robot or the user, etc. This will play a role in the use cases we describe in section 3.2 and 3.3 . The semantic structures consist of a dialogue act (intent) that adheres to the DIT++ schema and ISO 24617-2 standards [Bunt *et al.*, 2017], followed by the main frame/topic, taken from FrameNet where possible, and a number of (optional) key-value parameters. An example with slot fillers from the information state is shown here:

```
#YNQuestion(IsCloseBy, title={user.title},
  lastName={user.surname}, emotion=neutral)
```

The system outputs these dialog acts and sends them to a multimodal generation module, as well as to software modules of other partners, which may generate additional multimodal outputs, e.g., for an avatar.

Rule Language

The rule language of VOnDA was designed to be as compact and declarative as possible, but also allowing seamless integration of Java code and classes. One of the most important features is that RDF objects and classes can be treated similarly to Java objects, but additionally including *multiple inheritance* and *type inference*, which are built into the RDF/OWL semantics. The ontology that is part of every VOnDA project provides the necessary information for the compilation of the compact rule specifications into native Java code.[3]

To obtain modifiable objects in the RDF store, VOnDA does not delete or modify existing data, but only adds the updated information that carries the current time stamp. Internal background reasoning in the run-time system makes sure that when the rules are evaluated, the objects and “fields” (properties) are referenced, the current value is available, retaining the history of previous values.

3 Application Scenarios

By describing the current application scenarios we want to highlight the qualities of VOnDA that make it well-suited for implementing interactive systems, which are, among others:

[3]For a more in-depth discussion, see [Kiefer *et al.*, 2019]

- reliability and predictability of the resulting system
- real-time reactivity
- flexible dialogue modelling for complex dialogues with compact rule sets and reasonable implementation effort
- integration and use of heterogeneous data types, from sensor to semantic 2D map data
- an information state that is usable by all layers of processing: NLU, dialogue, action selection, NLG and multi-modal generation
- control and orchestration of major ensembles of complex modules and applications
- co-operation of modules by synchronised belief states or direct communication with exchangeable middlewares
- implementation of complex socially adequate agents exploiting the integrated long-term history

3.1 Command and Control for a Robotic Arm

One of the maybe least complex speech-based robotic applications is a verbal or multi-modal command and control interface for a robotic arm and its various capabilities.

The main challenge when implementing such an application is the high reactivity and reliability that is required to be able to control the arm safely. The latency between uttering a command and its execution has to be very small for some commands, e.g., for stopping a running command execution. Reliability in this case mostly depends on the accuracy of the automatic speech recognition, which can be tuned quite well towards accurateness despite the short utterances because of the small number of relevant possibilities. From the view of the dialogue and control management, latency and reliability are not an issue for a VOnDA implementation, since the rules can implement fully deterministic behaviour, and thanks to a synchronised information state between the dialogue system and the robot, the dialogue system is fully aware of the robot's internal state.

In the concrete case, we use a VOnDA dialogue system to control a robotic *Franka Emika* arm. The work is carried out as part of the INTUITIV project ([Kruijff-Korbayova *et al.*, 2020]), acronym for "Intuitive-nonverbal and informative-verbal robot-human communication", which aims at developing non-verbal and verbal human robot interactions that can be immediately understood by humans. The use-cases to study these interactions are applications in a rehabilitation clinic.

Here, the robotic arm is used to hand out required medical supplies to hospital staff during an examination session. The staff member moves it using speech commands in phases where both hands are already bound by other activities. More precisely, it directs the arm's end effector holding a tablet with medical consumables to (predefined) positions in space by means of voice instructions. The available guidance commands are shown in Figure 2.

The dialogue part of the system is responsible for translating the verbal commands into ROS[4] messages for the robot and generating speech output in case of errors or problems.

[4]http://wiki.ros.org/Documentation

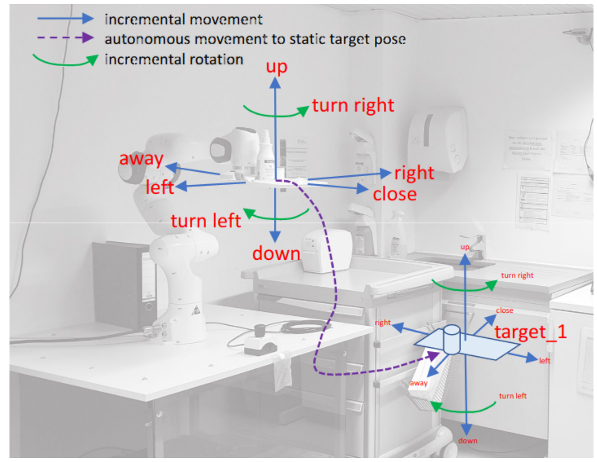

Figure 2: INTUITIV arm scenario: Support movement directions and target positions.

The robot performs the calculation of the specific actions necessary to reach the target position, while the VOnDA subsystem focuses on the speech interactions.

The information state contains concepts for the dialog history, dialog structure, history of commands, execution state of the robot as well as a list of predefined target poses/positions. For future applications, a facility to add poses and positions with the help of speech input is planned.

It is updated using two different input sources, firstly the intents determined by the language processing: Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) provided by Cerence Studio (former Mix)[5] platform, and secondly the execution status updates of the robotic system, e.g. ExecutionFinished_Success.

As described above, the rules for controlling the dialog translate the results of the language processing into ROS commands or generate language output if the execution status of the robot indicates an error. There are 4 main classes of rules:

- Directions, e.g. *right*, *left*, *up*, *down* etc.
- target positions, e.g. *go to target pose 2*
- Repeat/Cancel previous tasks, e.g. *continue*, *stop*
- Error messages, as reactions to robot execution status

All commands have instantaneous effects, which is especially important for the *Stop* command, which halts all activity of the arm, suspending it in the current position. Although an emergency button to halt the robot is still required, the voice command can probably be given faster than the mechanical signal.

3.2 Synchronised Multi-Modal Interaction Module for a Robotic Guide

The INTUITIV project also investigates how intentions of a robot can be made transparent to humans through anticipatory path selection in combination with iconic and verbal communication in order to minimise the discomfort due to uncertainty regarding the actions of a robot. The robotic platforms

[5]https://www.cerence.com/cerence-products/cerence-studio

used in this scenario are a guidance robot and a luggage transporter. Both robotic platforms are equipped with speech input and output as well as an stylised avatar face on a display.

This scenario poses quite different challenges to the underlying conversational system than the command-and-control scenario. Besides more complex interactions, which also have to be intuitive and compliant to the social interplay between robot and user, there are more real-time signals arriving from the robot in a higher frequency. Here, VOnDA makes it possible to react to these signals in real time, while at the same time providing a consistent and socially adequate interactive robot persona.

The robots picks up patients at some location within the clinic, such as the reception desk, and accompanies them to a destination such as the patient's room or a treatment/training facility. After receiving a pick-up order from the reception desk, the robot drives autonomously to the location of the patient and starts the guidance. During the guiding and autonomous phases of the robot, the main purpose of the VOnDA agent is the planning and execution of the situated dialogues, which consist of verbal as well as multi-modal output. The following main situations are covered by the system, with varied demands and requirements on the interaction management subsystem:

- *Greeting and Farewell:* The robot greets the client, introduces himself and states its current task. On a first meeting, the robot also describes how it works and how the client can interact with it and signals acquaintance otherwise. When reaching the destination, the robot politely says goodbye and returns autonomously to its starting position. Important here is that the interactions are adaptive, taking different forms depending on the level of familiarity between user and robot.
- *Navigation and guidance:* One of the main tasks of the robots is to guide patients through the clinic by giving them verbal (and multi-modal) route instructions, serving as an indoor navigation device. At first an (optimal) route based on the robot's position and destination on a semantic map of the building is computed, annotated with important landmarks and the functions of the rooms. During navigation, the robot's position data is compared continuously with the calculated path and outputs information about upcoming changes in direction etc. In addition to the guidance, the robot gives information about potentially interesting places on the way, and engages in other forms of social dialogue to make the journey more pleasant and less boring. This is facilitated in particular by VOnDA's information state containing the map information. Here, the flexibility of the RDF store to represent all kinds of data without the need of additional programming code is a big advantage. The RDF data for the map is mostly generated automatically using annotated OpenStreetMap data from the buildings in the *Indoor Mapping* format.
- *Episodic Encounters during autonomous drive:* An intuitive and user friendly system also implies that robots, even if they are autonomous, greet people in the corridor, make room for them, or otherwise interact with them, for example by announcing an overtaking manoeuvre or when the robot is addressed. The dialogue system uses a shared information state with the belief state of the robotic system to carry out such interactions, for example to find out if and how many people are around. The exchange with the robotic system is implemented via a ROS interface.
- *Using an Elevator:* During autonomous drive and when guiding the patient/client, it may become necessary for the robot to use an elevator. The use of an elevator poses a particular challenge to the robotic and the dialog system. The dialog required for this is very complex, since there are so many different situations and solutions possible, resulting in a wide spectrum of possible robot and user dialogue turns. This starts with asking for help when arriving at the elevator, since the robot can neither open the door by itself, nor can it press the button to bring it to the right level. In the following, appropriate reactions are needed if this help is granted or denied. In the following, the robot has to find a way to determine that it has arrived at the correct floor. This, again, can be done by asking people who are potentially present. If that is not the case, the robot might take up its planned journey until it can get evidence about which floor it is in, either with camera system support, map comparison, or asking people it encounters, if neither of the previous methods gives conclusive results (or is not available at all). Also, the situation with and without guided person might be quite different, and be even complicated by the fact that an elevator might be too small to accommodate the robot and the person. In situations like this, VOnDA allows very flexible dialogue flows that take a lot of parameters from the sensors and the information state into account.
- *Supporting the robot's recovery behaviour:* Exceptional situations, which occur both in the context of the autonomous journey and the journey with the patient, are solved in a collective effort between the robotic system and the dialogue system. For this purpose, there is a close cooperation between the components, which takes place via the ROS interface. The robotic system informs the dialog system about the nature of the problem, for example, the robot may not be able to continue its route because the corridor is blocked by a group of people. This information is used by VOnDA to launch a corresponding dialogue sequence. During the dialogue the system will send repeated requests and status updates to the robotic system. This way, the robotic system has knowledge of the dialog process and can initiate own actions if necessary, e.g. calculating an alternative route if the persons do not cooperate.

Although maybe not as advanced as the autonomous social robot from the MuMMER project, this is already a quite advanced use case, which highlights the abilities and advantages of a VOnDA based implementation.

In contrast to the robotic arm described above, the information state to be managed by the VOnDA system comprises a considerably higher number of information types from different sources. These include, among others, the semantic maps of the hospital, basic patient information, dialog history, representation of the different tasks and status and sensory information from the robots. The VOnDA agent maintains a distributed belief state, by synchronising the robotic knowledge state, e.g., robot status (idle, stuck, etc.), as well as the overall mission knowledge, with its internal information state relevant for dialogue, which is especially relevant for route guidance and recovery behaviour.

3.3 Central Control of a Robotic eHealth Teaching Companion

A fundamental goal of the PAL project[6] is to provide a platform for child health education (Type 1 Diabetes Mellitus), supported by an embodied robotic companion. After an assessment session to determine the child's physical situation and education needs, where a browser-based control panel is used to create an initial user profile, the children interact with a physical robot. First, it introduces itself and its purpose, and asks the child questions about him/her. Then, they play some rounds of educational games, which are selected by the child. Finally, a tablet with the PAL app is given to the children to accompany their learning process. A virtual avatar exactly simulating the robot (and the same software piloting it), is the central interactive component of the eHealth app on the tablet.

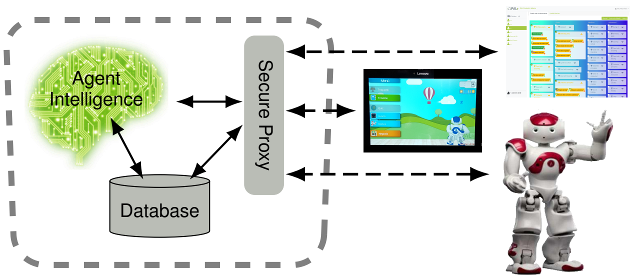

Figure 3: Architecture of the PAL system. External components are the robot, the tablet app and the patient control panel.

The *Hybrid Brain*

For every user login, a complete instance of the *"PAL brain"* is spawned, which consists of the modules shown in figure 4: dialogue manager with natural language understanding and generation (NLU/NLG), action selection/statistical user model, multimodal robot behaviour manager, targeted feedback, episodic memory, off-activity-talk, Explainable AI component. Many components are tightly linked do the dialogue manager that is predominantly implemented with VOnDA functionality, and is in fact the *central agent behaviour control component*, piloting the whole "brain" with the help of the rest of the modules and tying their functionality together.

The PAL platform is a cloud-based hybrid AI system [Kaptein *et al.*, to appear 2021], a complex mix of machine learning and symbolic AI components, implemented using different paradigms and programming languages. In the beginning, the hybrid approach allowed the first prototype of the system to be quickly available using more hand-crafted modules and settings. Then, the system gradually evolved to using more and more learned and sophisticated components once the data became available.

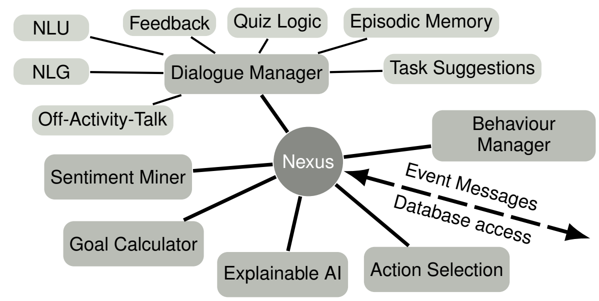

Figure 4: The "Hybrid Brain" of the PAL system

A single RDF database (a HFC storage/reasoner) instance provides the central cloud data hub for all subsystems. Most of the communication between modules and components, including that of the (partly virtual) robotic agents go through this hub. In addition, modules of one "brain" instance can exchange direct messages using a communication layer implemented by Apache Thrift (https://thrift.apache.org/).

The RDF storage contains an extensive ontology of health and learning related information, as well as sub-ontologies for human/machine roles or tasks, events, processes, natural language semantics, episodic memory structures to capture specific events, an off-activity database for social talk, etc. The heterogeneous knowledge and data in the database is shared among all components, and used in such diverse tasks as action selection, the robot behaviour management, which includes motions and utterances, interaction management, such as task suggestions, social talk, relating to past episodes, and all higher-level applications. Almost all components use the database in both ways, for gathering as well as providing data.

Activities in the eHealth App

The child can interact with her/his robotic pal in several activities which are components of the app, and make quite different demands on the central control module:

Educational Games

Quiz, Memory Games, Break and Sort, Diabetes Notebook

Fun activities

Robotic Dance Creator, Add-On Shop (gamification)

Conversational activities

Task suggestions and explanations, Achievement tracking, Off-activity talk, immediate progress feedback and mid-term aggregated feedback through the episodic memory module

There are many demanding aspects in these activities that the implementation with VOnDA is able to support:

The already large number of components and apps, many influencing each other in their behaviour by generating new data about the user, needs to be arbitrated. This is complicated by the fact that the user can spontaneously stop any activity at almost every point in time or switch between them, and the agent has to consistently react to such spontaneous user behaviour. The pseudo-parallel nature of the VOnDA

[6] https://www.pal4u.eu

rule system (all rules may be triggered more or less simultaneously) is a much bigger advantage than the potential problems which may arise by such an architecture. The size of the rule system implementing the PAL system could be kept quite small, using rules with general preconditions that are equally applicable in many situations, and since all rules are executed on information state changes unless special measures are taken, rules are in principle "always on". On the other hand, it is also easy to fence certain rule subsystems from others by applying "bail-out" rules based on the current system state. In fact, the core control component with all natural language resources was build in about half a person-year, parallel to the VOnDA framework.

To allow for variability and spontaneity in the reactions, alternative action paths are added at all places where it is sensible in the interaction flow. The choice is left to the data-driven statistical action selection module, which takes the current information state into account to take the most appropriate choice at the given situation. This choice is highly adaptive also to the user model and perceived user state, e.g, the current game or achievement progress.

While the real-time data frequency (game scores, user answers, data entry) is on the medium to low level, the reactivity of VOnDA allowed fine grained synchronisation with the all tablet components and the robot, e.g., to guarantee that robot movements and utterances don't get into each other's way, game results and tablet input are immediately passed between tablet and the central component for proper reactions, etc. Additionally, the multimodal generation modulates the robot gestures and utterances using knowledge stored in the user profile of the information state.

Having all interaction history available allows different forms of long and short-term interaction patterns: immediate game and achievement progress, mid-term reactions using aggregated data in the episodic memory, machine-learning based user modelling for adaptivity of tasks suggestions, and bonding effects using interactions with the gamification incentives and the off-activity talk.

4 Evaluation of System Use

We present some numbers from the PAL system's last experimentation cycle. The other projects are still in progress and have unfortunately not produced enough data for decisive results.

In this experiment, children interacted with the continuously running system over an extended period of time (48 children from Italy and the Netherlands over 2.5 to 3 months), in 571 resp. 471 sessions. Table1 lists figures indicating usage patterns per child for both countries: Number of sessions, activities during one session, and average session and activity duration in seconds. While children seem to use the system quite differently, which is visible in the high variation in usage duration, usage across countries looks rather similar. Unfortunately, we have no in-depth evaluation about usage patterns in different child populations. The robot addressed the child about 38 times in every session, from a repertoire of around 4800 unique utterances per language. Repetitive in-game feedback is not taken into account here. The children addressed the robot using free-text input around ten times per session. Most of the interaction could be handled using App buttons or other GUI elements.

	# sessions	duration	# activities	act. duration
IT	17.3(14.5)	819.8(1018.1)	14.0(30.0)	58.4(159.8)
NL	16.8(13.2)	721.5(939.2)	12.6(17.7)	57.1(154.7)

Table 1: Average usage parameters per child: standard deviations are in parentheses

5 Future Work

While there are already tools for editing, browsing and debugging the rule system present in the current framework, having additional and more sophisticated tools almost always helps for implementing system, and makes it more pleasant to work with it. We are currently adding a graphical editor for hierarchical state charts, which is an easy to grasp format for beginners, and a compiler that transfers these state charts into VOnDA rules. The goal is to give new developers an easy head start and at the same time be able to tweak the result (the compiled automaton) with the much more powerful and better generalising reactive rule framework.

There are also plans to make the syntax easier to read for the developer *and* the compiler, since some language constructs of Java that are currently supported are unnecessarily hard to parse.

6 Conclusion

We believe to have demonstrated that the VOnDA framework is feasible for implementing a significant range of robotic conversational systems. Its compact specification language, seamless integration with a programming language, and connection to the extended RDF reasoner HFC open many ways and opportunities for exploitation.

The PAL system gives proof that stable and reliable systems can be build in this way. Although the server hardware was very moderate, no performance problems were reported, even when 40 users were logged in at the same time. The server also did not have to be restarted at all in the final trial period, which lasted 3 months.

A bigger developer base would provide the best hints as to in which direction to develop the framework, and we still believe that hybrid AI has many benefits over pure data driven approaches, at least for a significant range of applications, so we hope to have drawn in more adopters with this paper.

Acknowledgements We want to thank all the collaborators in the projects that helped us implementing these systems. Special thanks go to Ivana Kruijff-Korbayová for her valuable comments and suggestions, and to Bert Bierman for constant mental support.

The research described in this paper has been funded by the Horizon 2020 Framework Programme of the European Union within the project PAL (Personal Assistant for healthy Lifestyle), Grant agreement no. 643783, and the BMBF project INTUITIV, Grant No. 16SV7979.

References

[ASAROB Team, 2020] ASAROB Team. ASAROB project website. https://asarob.de/, 2020.

[Belpaeme *et al.*, 2013] T. Belpaeme, P. Baxter, R. Read, R. Wood, H. Cuayáhuitl, B. Kiefer, S. Racioppa, I. Kruijff-Korbayová, G. Athanasopoulos, V. Enescu, R. Looije, M. Neerincx, Y. Demiris, R. Ros-Espinoza, A. Beck, L. Cañamero, A. Hiolle, M. Lewis, I. Baroni, M. Nalin, P. Cosi, G. Paci, F. Tesser, G. Sommavilla, and R. Humbert. Multimodal Child-Robot Interaction: Building Social Bonds. *Journal of Human-Robot Interaction*, 1(2):33–53, 2013.

[Belpaeme *et al.*, 2015] Tony Belpaeme, James Kennedy, Paul Baxter, Paul Vogt, Emiel EJ Krahmer, Stefan Kopp, Kirsten Bergmann, Paul Leseman, Aylin C Küntay, Tilbe Göksun, et al. L2TOR-second language tutoring using social robots. In *Proceedings of the ICSR 2015 WONDER Workshop*, 2015.

[Bunt *et al.*, 2017] Harry Bunt, Volha Petukhova, David Traum, and Jan Alexandersson. *Dialogue Act Annotation with the ISO 24617-2 Standard*, pages 109–135. 11 2017.

[Foster *et al.*, 2016] Mary Ellen Foster, Rachid Alami, Olli Gestranius, Oliver Lemon, Marketta Niemelä, Jean-Marc Odobez, and Amit Kumar Pandey. The MuMMER project: Engaging human-robot interaction in real-world public spaces. In *International Conference on Social Robotics*, pages 753–763. Springer, 2016.

[Foster *et al.*, 2019] Mary Ellen Foster, Bart Craenen, Amol Deshmukh, Oliver Lemon, Emanuele Bastianelli, Christian Dondrup, Ioannis Papaioannou, Andrea Vanzo, Jean-Marc Odobez, Olivier Canévet, et al. MuMMER: Socially intelligent human-robot interaction in public spaces. *arXiv preprint arXiv:1909.06749*, 2019.

[Kaptein *et al.*, to appear 2021] Frank Kaptein, Bernd Kiefer, Antoine Cully, Oya Celiktutan, Bert Bierman, Rifca Peters, Joost Broekens, Willeke van Vught, Michael van Beckum, Yiannis Demiris, and Mark A. Neerincx. A cloud-based robot system for long-term interaction: Principles, implementation, lessons learned. *Transactions on Human-Robot Interaction*, page 17, to appear (2021).

[Kennington *et al.*, 2020] Casey Kennington, Daniele Moro, Lucas Marchand, Jake Carns, and David McNeill. rrSDS: Towards a robot-ready spoken dialogue system. In *Proceedings of the 21th annual meeting of the special interest group on discourse and dialogue*, page 132–135, 2020. Citation Key: kennington2020rrsds.

[Kiefer *et al.*, 2019] Bernd Kiefer, Anna Welker, and Christophe Biwer. VOnDA: A Framework for Ontology-Based Dialogue Management. In *International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, page 12. Springer, 4 2019.

[Krieger, 2013] Hans-Ulrich Krieger. An efficient implementation of equivalence relations in OWL via rule and query rewriting. In *Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on*, pages 260–263. IEEE, 2013.

[Krieger, 2014] Hans-Ulrich Krieger. A detailed comparison of seven approaches for the annotation of time-dependent factual knowledge in rdf and owl. In *Proceedings 10th Joint ISO-ACL SIGSEM Workshop on Interoperable Semantic Annotation*, 2014.

[Kruijff-Korbayova *et al.*, 2020] Ivana Kruijff-Korbayova, Johannes Hackbarth, Caspar Jacob, Bernd Kiefer, Matthias Schmitt, Tanja Schneeberger, Tim Schwartz, Hanns-Peter Horn, and Karsten Bohlmann. Towards Intuitive Verbal and Non-Verbal Communication for Incidental Robot-Human Encounters in Clinic Hallways. In *International Conference on Human-Robot Interaction 2020*, 2020.

[Lison and Kennington, 2016] Pierre Lison and Casey Kennington. OpenDial: A toolkit for developing spoken dialogue systems with probabilistic rules. *Proceedings of ACL-2016 System Demonstrations*, page 67–72, 2016.

[Mocan *et al.*, 2016] Bogdan Mocan, Mircea Fulea, and Stelian Brad. Designing a multimodal human-robot interaction interface for an industrial robot. In *Advances in Robot Design and Intelligent Control*, pages 255–263. Springer, 2016.

[PAL, 2018] PAL. The PAL Project Website. http://www.pal4u.eu/, 2018.

[Ramgire and Jagdale, 2016] J. B. Ramgire and S. M. Jagdale. Speech control pick and place robotic arm with flexiforce sensor. In *2016 International Conference on Inventive Computation Technologies (ICICT)*, volume 2, pages 1–5, 2016.

[SynDiQuAss Team, 2020] SynDiQuAss Team. SynDiQuAss project website. https://syndiquass.de/, 2020.