ROBUST ADAPTABLE HUMAN-ROBOT DIALOGUES FOR PRODUCTION PROCESSES

Tobias Theuerkauff [*] **Yves Wagner** [*] **Jan P. Vox** [*]
Karen Insa Wolf [**] **Frank Wallhoff** [*,**]

[*] *Institute for Technical Assistive Systems, Jade University of Applied Sciences, Ofener Str. 16/19, 26121 Oldenburg (e-mail: {tobias.theuerkauff,yves.wagner,jan.vox, frank.wallhoff}@jade-hs.de).*
[**] *Fraunhofer Institute for Digital Media Technology IDMT, Marie-Curie Straße 2, 26129 Oldenburg (e-mail: {insa.wolf,frank.wallhoff}@idmt.fraunhofer.de)*

Abstract:
Future industry is characterized by high product individualization. Individualization also means the need for flexible production. In the future, people and robots will work increasingly closely together into order to maintain flexibility. The modeling and implementation of an adaptive assistance robot is therefore aimed at in this project. The robot shall exclusively be controlled by voice allowing for an hands-free production. The system should be able to learn new scenarios and individual working steps, in order to carry them out as required. The final goal of the project is the exploration of interaction strategies between a human worker and a robot based on novel dialog models.

Keywords:
Adaptive algorithms, Adaptive control, Adaptive Systems, Cognitive interaction, Computer-aided work, Human-machine Interface, Industrial control, Industrial production systems, Industrial robots, Speech control

1. INTRODUCTION

Because for safety reasons in the nowadays production processes people and machines operate in a separated manner, there is so far only little knowledge of how collaborative processes between robots and humans have to be designed. One approach is to provide the robot with an adaptive dialogue system which, depending on the possibly incomplete inputs of the human being, can produce a suitable reaction, e.g. the reaching of tools or the retrieval of components. Since in industrial production, The hands of the worker are not available for haptic input through activities or tools, language control would be an alternative.

This project deals with the question how a dialogue structure of an assistance system must be built in order to use an industrial robot in a collaborative manner. The goal is the creation of a functional and learnable demonstrator, which is controlled by voice commands and simultaneously captures its environment via cameras.

2. RELATED WORKS

According to Thiemermann (2005), the cooperative work between humans and robots can be divided into the three types autarkic, cooperative and synchronized working. When they are work in an autaric area, there is a local separation between the worker and the robot, at the synchronized working type there is a temporal separation during the work. The third kind, the cooperative type, allows human beings and robots to complement each other and to support them according to their specific abilities. For optimal cooperation between humans and the robot, it is an advantage if they can communicate and coordinate with each other. Such a cognitive system is described in Bubb et al. (2014) in this setting the worker is supposed to be in a manual assembly context. A robot should provide the human with necessary information at the right time. Assistive information is displayed to the worker on a touch screen. Thus the worker can communicate with the robot via touch input. In Bannat (2014), a test system for an assembly scheduling assistant was imbedded. A model consisting of several groups of assembly groups should be mounted by the user using the assistance system. As soon as the worker has grasped a component, all parts belonging to that group were automatically displayed to him, including an assembly instruction. Step by step, the model could be assembled without prior knowledge by the user. In addition to touch gestures and visually observed actions, a further possibility to provide a human-robot-cooperation can be achieved via voice communication. According to a survey presented in Berg (2014), a large proportion of the respondents were convinced that Spoken Dialogue Systems (SDS) are already beneficial or will be beneficial in the future. But, SDS can have the problem, that a statement of the user is not recognized. In the

worst case, a program would be terminated. In Henderson et al. (2012) the topic is examined as the system can find from a non-understanding error back into the dialog. The project has developed different strategies that allows the SDS to move in a different direction in the case of a non-understanding error. Thus the program is not terminated unexpectedly.

3. CONCEPT

In an industrial test environment, a user shall be supported by a robot while building assembly groups. These groups consist of several sub-assembly components, which are handed over to the robot in small boxs. For this purpose, a new design sequence for a model must first be introduced to the robot. Afterwards, the user can teach the robot the course of the construction of assembly groups by transferring new components to the robot (learning scenario). After a box is transferred, users are asked for their contents. The assistance system associates the contents of the box together with the QR code of the box in a database. Processed components are finally placed it finally in a free box slot on the working table. Furthermore, the system stores the order in which the boxs are handed over. If the user wants to re-build a model after the learning process, he can recall the saved scenario. The robot supplies the required components to the user in the correct order. The robot should also check whether the required components are available for assembly. If this is not the case, an appropriate interaction with the user must be started in order to change the scenario or to add the missing components. The transfer of the components as well as the learning and execution of scenarios is to be carried out exclusively by voice control. A further system feature in this project shall be, that an initial adaptation of the user to the system may be omitted. The prerequisite for this requirement arises from the fact, that the robot communicates interactively with the user. In the case of availability problems (eg. missing components), the program is not expected to abort the assembly, but to assist the user to solve the problem (eg. introducing a new component) via voice communication. At any time, the system should adapt to the human being and have an individual learning effect.

3.1 DEMONSTRATOR COMPOSITION

For the prototypes presented in this project, the environmental structure, described in the following, was developed. An overhead camera is installed on the ceiling over the table on which the roboter (KUKA KR3) is mounted (Fig. 1). The camera is used for the general overview of the table (totalview) and for the position determination of the individual boxs. Another camera is attached to the gripper of the robot arm (tool-center-point camera). This one is used for precise navigation of the gripping arm to the boxs as well as for reading the QR codes. The QR codes on the boxs are necessary in order to clearly distinguish them.

3.2 DEPLOYED HARDWARE

The model KR3 from KUKA was used as worker assistance robot (Fig. 2). The KR3 model is a six-axis industrial robot with a maximum trailing load of 3kg.

Fig. 1. Construction of the test environment with the robot and the cameras

Fig. 2. The KUKA KR3 in the test environment

At the end of its arm (tool-center-point (TCP)) a hydraulic gripper from FESTO is mounted, which can be closed and opened by means of a compressor. The robot is permanently mounted on a table and operates in an TCP action radius of max. 635mm (plus the dimensions of the gripper). The positioning of the arm takes place via Kartesian coordinates, an axis-specific control via angle indications is also possible. The origin of the coordinate system is located in the base of the robot as depicted in Fig. 3.

The robot is programmed with the programming language KUKA KRL (Kuka Robot Language). This is a script language similar to BASIC. The program is executed on the KR C2 (Kuka Robot Controller). The KR C2 contains the entire power electronics for the robot as well as a PC (Windows XP Embedded) for executing the programs. The communication to the outside is done by a serial interface (COM port) controlled by an internal state machine avoiding the navigation to critical positions.

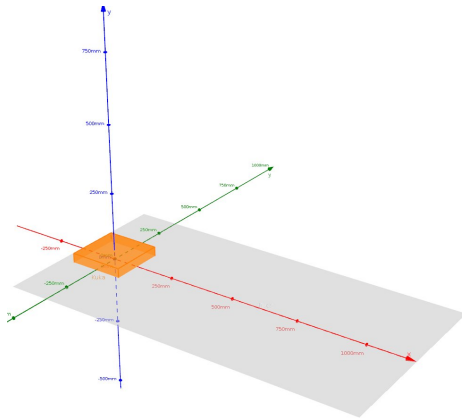


Fig. 3. Coordinate system of the KUKA robot. Schematic representation of the base of the KR3 (orange) as well as the table (gray)

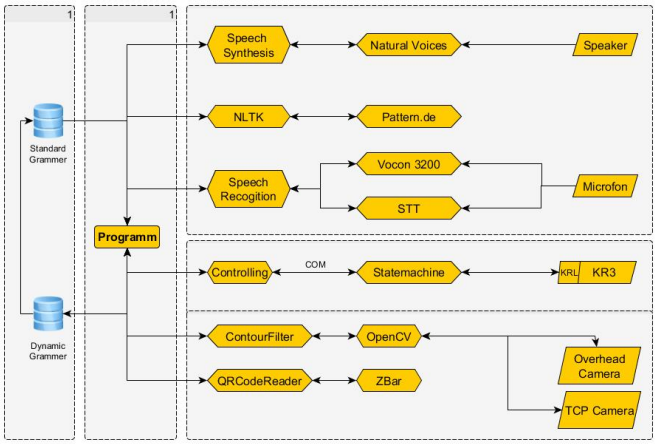


Fig. 4. Software architecture of the prototype

4. SOFTWARE IMPLEMENTAION

To meet the requirements set in (3), two main components must be developed. On the one hand, a dialog structure must be designed that allows the robot to communicate with the user. The dialog structure must be designed in such a way that it can be expanded dynamically during the work process. Unknown terms (learning new parts and scenarios) must be incorporated into the grammar at runtime. The speech recognition must also function reliably in the vicinity of machine noises.

On the other hand, image processing must be implemented, which is capable of evaluating video in real time. This part is required since the robot must also interact with the user. The individual components have to be delivered by the robot to the user. Likewise, the user must be able to transfer boxs to the robot when introducing new components

4.1 DEVELOPED AND DEPLOYED SOFTWARE

Our project covers several areas of data processing, as described in chapter (1). The individual components are interdependent, as can be seen in (4). The various system components are explained in more detail below.

SPEECH PROCESSING The relevant functional aspects of speech analysis and speech synthesis will be recapitulated first. Most of the speech analysis and synthesis modules is carried out with the software DialogOS of *CTL Sprachtechnologie*. DialogOS is a graphical development environment for developing dialogue systems. In the background of the development environment DialogOS uses the speech synthesis product Natural Voices from *AT&T* as well as *VoCon3200* from Nuance for speech recognition. Grammar can be defined in DialogOS that define the complexity of the speech recognizer output. These grammars consist of individual or composite rules. Depending on which rule applies, a return value (so-called tags) can be defined and further processed. This interface is used in the project to generate external grammars and to be processed by DialogOS (CLT Sprachtechnologie GmbH i. L. (2016); Wichert and Klausing (2013)).

DialogOS allows speech recognition by means of a pattern with a discrete vocabulary of words. That means, the possible statements of the user must be known in advance and defined in rules. On the other hand use of context-free-grammars allows for highly controlled return values even in noisy environments due to a adjustable complexity of the grammar. As soon as an word that is not yet part of the vocabulary set is expected, a second *speech recognition* module is used. It can convert any spoken words to text (Speech To Text (STT)). Speech recognition uses various engines for analysis (e.g., *CMU Sphinx* or *IBM Speech to Text*). With an active connection to the Internet, it is also possible to link Google's Speech Recognition API. The software *Pattern* of the research center *CLiPS* was additionally used to keep the dialogues more flexible. It is a program for the analysis of natural language. For example, it is possible to split sentences into individual words (Part-of-Speech-Tagger). The software also controls the conjugation of verbs and the singularizing / pluralizing of nouns. In this application, the singularizing / pluralizing of nouns is mainly used to generate grammars for the use with DialogOS (Anthony Zhang (2016)).

DYNAMIC DIALOUGE GENERATION The grammars for DialogOS are generated during the training phases (learning of new components or scenarios) in the running application. The interaction grammars are not implemented in DialogOS at the beginning, but are generated dynamically at run-time by applying the steps presented above, and are interfaced by DialogOS afterwards.

At the beginning, the grammars contain only rules for selecting basic functions. These basic functions include, for example, the rules for adding new components or scenarios. The basic grammar is depicted in Fig.(B). The individual rules can contain placeholders ($name), which in turn are replaced by extended rules (Fig.(C)). However, such basic functionalities are not sufficient for the requirements placed on this project, as the scenarios described below intentionally illustrate.

The user can ask the robot to include new components in his set of components. During this interaction process, the user is asked which component is involved. At this moment, the speech analysis is excluded via DialogOS, since no comparison patterns can be generated. Accordingly, extended speech recognition has to be used, since the

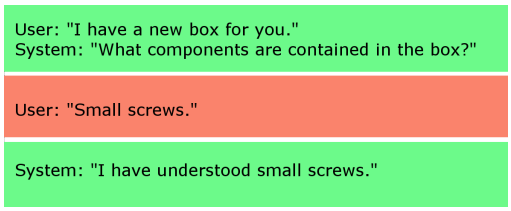


Fig. 5. Example dialog with the change of the interpretation software translated into English; green: DialogOS, red: STT

grammar has to be expanded dynamically by new terms (name of the component). For this purpose, the user's statement is converted into text within this project via the *Google Speech Recognition API (STT)*. The new term is stored in a database with the QR code on the box as well as the storage position of the box. To expand the grammar, it is necessary that the robot prompt the user to explicitly name the new component (Fig.(5)).

After applying the STT to the spoken statement of the user, the name of the new component is presented as text and can be integrated into the grammars for the DialogOs software as well as it is stored in the database with the position and content data of the associated box to be stored. In this process, the grammar for DialogOS is expanded dynamically by the contents of the box. In the generation of the grammar, patterns are also generated which allow the contents of the box to be named in the sigular or plural form (stemming). The reason for this is that the user, for example, says "screws" (plural) but then only wants to have a "screw" (singular) ("Please give a screw.") . Thus patterns are created, which apply to both forms (singular / plural) and can be recognized by DialogOS. These patterns are generated by *CLiPS* with the *pattern* software presented in Fig.(4.1.1). This allows the user to request the new component in the next pass without having to pay attention to a special way of expressing the name of the component.

The process of learning new scenarios is based on a similar principle. *Google's Speech Recognition API* is also used to name a new scenario. The learning process for a new scenario is displayed in Fig.(A) via a state diagram. The dialog was translated into the English language for a better understanding. The actual implementation took place in German as seen in Fig.(B) and Fig.(C). During the learning process the grammars are extended by the name of the new scenario. After creating a new scenario, the required components can be iteratively learned, as described above. The order in which the components are introduced is also stored in the database along with the QR code of the associated box as well as the storage position of the box. Through this process, the handling process for producing a model can be completely reconstructed by the software. The user can ultimately re-call a previously learned scenario and gets all needed parts for the model in the right sequence from the robot.

IMAGE PROCESSING The second observation modality besides speech, included in our project, consists of image processing. As already described in Fig.3, small boxes are used for the individual components. These boxes have to be detected by the robot. ***OpenCV*** (Open Source Computer Vision) is a C / C ++ library for real-time processing of image information. Face and gesture recognition, contour or object recognition as well as various filters are a small part of the functions of the library. OpenCV is available for almost all operating systems (Windows / Linux / Mac / Android / iOS) (Bradski (2000)). In order to decode the QR codes of the boxs, the Open Source project ***zBar*** is used. With the software developed there, it is possible to decode bar codes or QR codes from images (Jeff Brown (2011)). In addition, a C library is provided, which can be integrated into own projects.

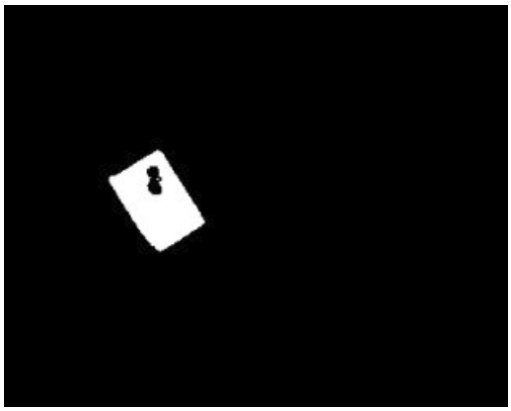


Fig. 6. After the applied color filter, the box can be clearly recognized in the intermediate result

In order to accept a new box from the user or transfer a box to the user, image processing algorithms are used. This method is necessary so that the robot can precisely position the gripper arm at the position of the box at which shall be grabbed and lifted. Two basic operations are necessary for this operation. First, the coarse position of the box on the table is determined using the overhead camera Fig.(7).

In the second step, the gripper is navigated to the coordinates of the gripping point on the box using the tool-center-point camera. Both base operations consist, in turn, of several individual steps from the field of image processing, which are solved using the OpenCV library. For a simplified subsequent processing of the image data, images are first converted from the RGB color model (red-green-blue) into the HSV color model (Hue-Saturation-Value). The HSV color model allows one color to be expressed by a single value, thus simplifying the confinement of the colour range for the filter (Gonzalez and Woods (2002)). Next, the image is smoothed by noise suppression and converted into a black-and-white image using a colour filter Fig.(6).

From the resulting image, the contours of the box are calculated by means of a contourfilter of the OpenCV library. The minimal surrounding rectangle (MSR) of the box can then be determined by means of a further function. From the coordinates of the MSR obtained in this way, the two short sides of the box are determined and their center points are calculated since these two points are used as potential gripping points for the robot. To determine the gripping point to be used, the point closest to the robot must be calculated. In the last step, this point is transformed from the image coordinate system into the

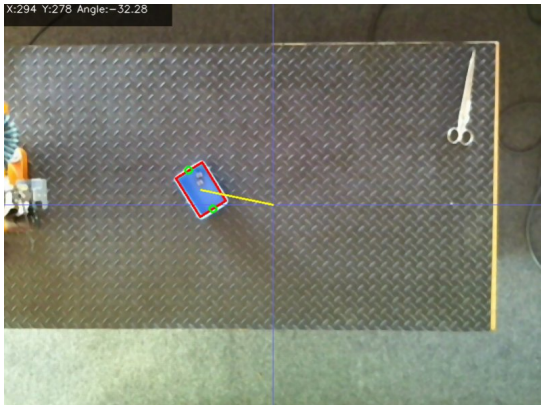
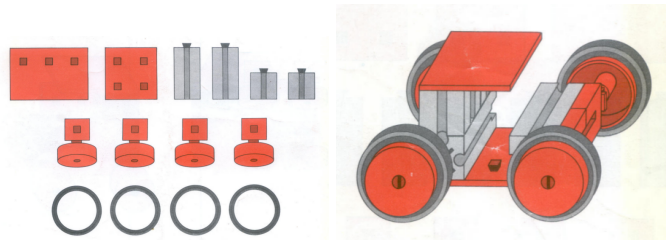


Fig. 7. Result of box recognition. The red square shows the box, the green points the gripping points of the KUKA.



(a) The individual components of the test model (b) The fully assembled model

Fig. 8. Testmodel (Taken from the construction manual Small Car, ©Fischertechnik)

coordinate system of the robot and passed over as the final target point to the control software.

As soon as the QR code of the controlled box is visible in the sequence described above, this is also evaluated. The robot stores the QR code information (box number) and links it later in the database with the contents of the box that the user has entered.

EXAMPLE SCENARIO The test scenario was the same assembly of a simple car from six different *Fischertechnik* components as in Bannat (2014). As assembly process, a new scenario was created using the speech control. The introduction of the individual (new) components were exclusively controlled by voice, transferred in the order of the assembly sequence (Fig.(4.1.4) (a)). After the subsequent execution of the learned test scenario, the user of the system receives a ready-assembled car, which he could assemble in a collaborative assembly process with the robot (Fig. (4.1.4) (b)) again.

5. CONCLUSION AND OUTLOCK

Within the project, a fully functional and adaptive system for the cooperative assembly between a human worker and a robot have been developed. The system is able to dynamically learn, manage and assist during the construction during various scenarios with different assembly sequences. Empirically, the functionality was tested for the construction of a simple car model from Fischertechnik components. In the future, the demonstrator will be used for further research in the area of human-robot cooperation. Specifically, the created demonstrator is currently being ported to the new robot platform *YouBot*. In the next step the demonstrator will be extended with optical devices to capture user motions. These provides the ability to control and interact with the robot via gestures. To achieve this, the Microsoft ™ Kinect v2 will be integrated in the demonstrator. The Kinect v2 and the associated API provide a opportunity to recognize Cartesian joint XYZ-positions in 3D space. In addition the authors are envisionig the correlation of a user's position with position data from the robot in a common space. The objective is to optimize the collaboration and to prevent collision between the user and the robot. The collaboration with the new YouBot demonstrator setup including the Kinect v2 is shown in Fig.(9). Implicit learning using visual inspection exting the speech interaction is also a demanding task, that will be considered in the future. A further challenging research question is wether the efficiency and ergonomics of a workplace can be increased by means of a hybrid assembly system in the context of the research field *Industry 4.0*.

Fig. 9. Use of the developed software on the robot YouBot from Kuka

ACKNOWLEDGEMENTS

The authors received grants from the internal research and development funds from Jade University of Applied Sciences.

REFERENCES

Anthony Zhang (2016). Speech Recognition (Version 3.4). URL `https://github.com/Uberi/speech_recognition`.

Bannat, A. (2014). *Ein Assistenzsystem zur digitalen Werker-Unterstützung in der industriellen Produktion*. Ph.D. thesis, München, Technische Universität München, Diss., 2014.

Berg, M. (2014). Survey on spoken dialogue systems: User expectations regarding style and usability. *XIV International PhD Workshop OWD 2012, 20–23 October 2012*.

Bradski, G. (2000). The OpenCV Library.

Bubb, H., Müller, P.D.H., Schubö, D.A., habil. G. Rigoll, P.D.I., Wallhoff, D.I.F., , and Zäh, P.D.I.M.F. (2014).

Cotesys progress acipe - adaptive cognitive interaction in production environments. *DFG Cluster of Excellence: Cognition of Technical Systems*.

CLT Sprachtechnologie GmbH i. L. (2016). *CLT Sprachtechnologie - DialogOS Übersicht*. URL `http://www.clt-st.de/produkte-losungen/dialogos/ubersicht/`.

Gonzalez, R.C. and Woods, R.E. (2002). *Digital image processing*. Prentice Hall, Upper Saddle River, N.J, 2nd ed edition.

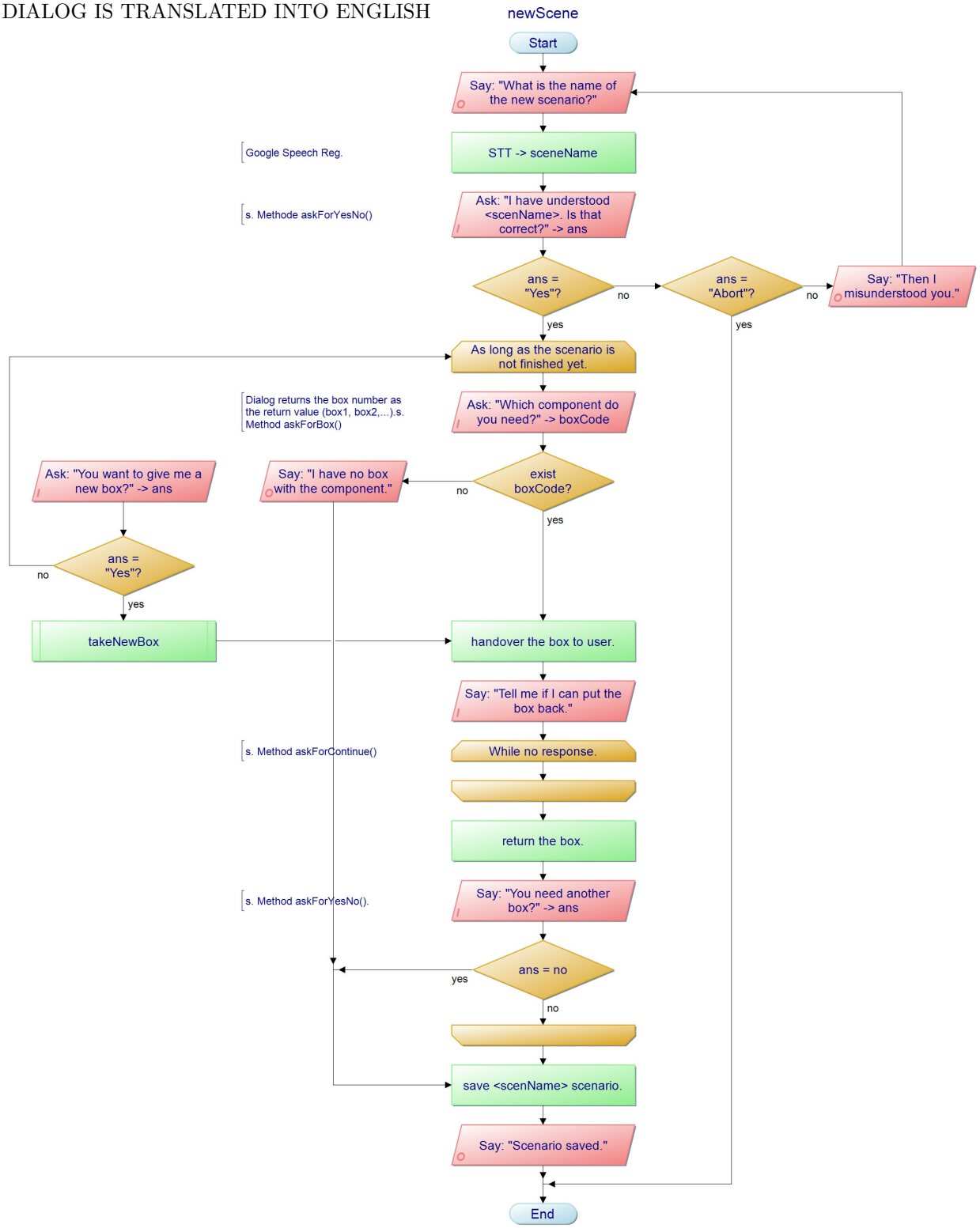
Henderson, M., Matheson, C., and Oberlander, J. (2012). Recovering from non-understanding errors in a conversational dialogue system. In *Proceedings of SemDial 2012 (SeineDial): The 16th Workshop on the Semantics and Pragmatics of Dialogue*, 128. Citeseer.

Jeff Brown (2011). *ZBar bar code reader*. URL `http://zbar.sourceforge.net/`.

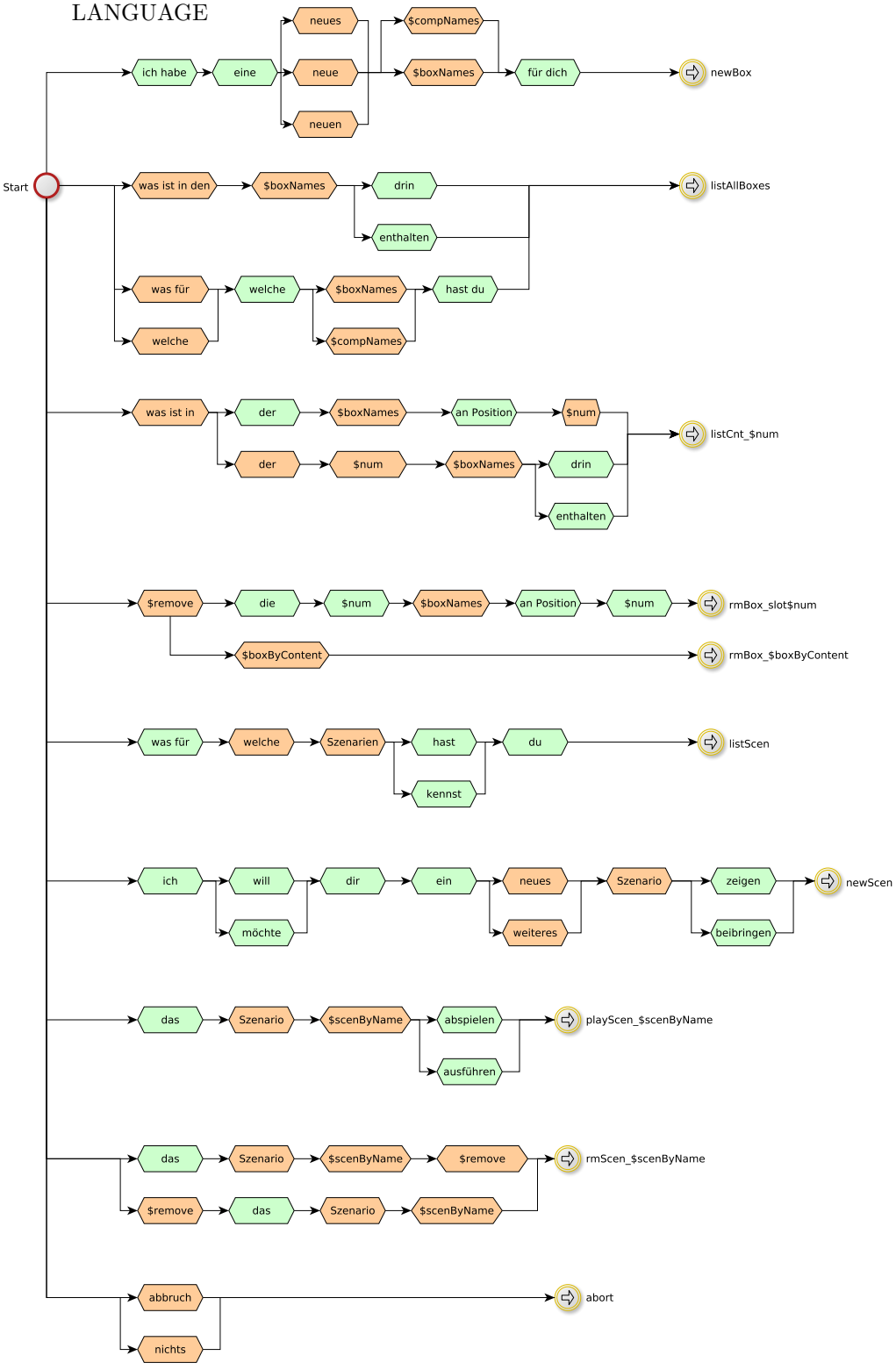
Thiemermann, S. (2005). *Direkte Mensch-Roboter-Kooperation in der Kleinteilemontage mit einem SCARA-Roboter*.

Wichert, R. and Klausing, H. (2013). *Ambient Assisted Living: 6. AAL-Kongress 2013 Berlin, Germany, January 22. - 23. , 2013*. Springer Science & Business Media.

Appendix A. TEACHING A NEW SCENARIO DEPICTED IN A FINITE STATE AUTOMATA. DIALOG IS TRANSLATED INTO ENGLISH



Appendix B. AN OVERVIEW OF THE STARTUP GRAMMAR IN THE ORIGINAL GERMAN LANGUAGE



Appendix C. AN OVERVIEW OF THE RULES USED IN THE PLACEHOLDERS IN THE ORIGINAL GERMAN LANGUAGE