Evaluating Knowledge-Based Assistance for DIY

Marvin Schiller[1], Gregor Behnke[1], Pascal Bercher[1], Matthias Kraus[2], Michael Dorna[3], Felix Richter[3], Susanne Biundo[1], Birte Glimm[1], Wolfgang Minker[2]

Institute of Artificial Intelligence, Ulm University, Germany[1]
Institute of Communications Engineering, Ulm University, Germany[2]
Corporate Research, Robert Bosch GmbH, Renningen, Germany[3]

marvin.schiller@uni-ulm.de, gregor.behnke@uni-ulm.de, pascal.bercher@uni-ulm.de, matthias.kraus@uni-ulm.de, michael.dorna@de.bosch.com, felixmilo.richter@de.bosch.com, susanne.biundo@uni-ulm.de, birte.glimm@uni-ulm.de, wolfgang.minker@uni-ulm.de

Abstract

We report on the development of a companion system incorporating hierarchical planning, ontology-based knowledge modeling and multimodal cloud-based dialog. As an application scenario, we consider the domain of do-it-yourself (DIY) home improvement involving the use of power tools. To test and – if necessary – adjust the developed techniques, user studies are conducted throughout the development phase. We present fundamental considerations and open questions encountered when testing the implemented prototype with potential users and report first observations from a current study.

1 Introduction

Traditional forms of instruction, such as user manuals or tutorial videos, are inherently limited by their format. They are not suited to take into account a great variety of circumstances a user might be in, and cannot cater for individual levels of prior knowledge. To address such limitations, we started the development of a companion system prototype using hierarchical task network (HTN) planning and ontology-based knowledge modeling (Behnke, Schiller, et al., 2018). We apply this prototype to assist novice users in the do-it-yourself (DIY) home improvement domain, where instruction can potentially be varied to take into account:

1. A user's prior knowledge and proficiency in dealing with electric and manual tools
2. A user's general knowledge about the employed materials and techniques
3. The availability of tools, attachments and materials
4. Different possible courses of action towards a goal
5. Unforeseen events during task execution, e.g. if a tool breaks or a mistake is made

To achieve the desired flexibility, our approach incorporates a full-fledged planning system in the form of a (SAT-based) HTN planner (Behnke, Höller, et al., 2018), an ontology for factual background knowledge on materials and tools, and a multimodal dialog-based user interface. In a complex problem-solving scenario such as in the DIY domain, hierarchical planning enables the presentation of instructions at different levels of granularity, by exploiting the hierarchical levels at which tasks and their respective subtasks are modeled. Our modeling approach combines planning with ontology reasoning and supports the user both at the procedural and the conceptual level: by demonstrating the use of tools, by planning sequences of steps and substeps to achieve a certain goal, and by providing useful information about tools and materials that allows the users to gain expertise while working with the system. A further level of flexibility is offered to the user in the form of multiple input modalities; the user may employ traditional click and touch gestures to operate the interface, but can also ask questions or issue commands using either speech input or typed text. Having potential users interact with this flexible system in a series of experiments allows us to study the interaction and to identify potential problems.

The idea of interactive user assistance has been picked by related work in the context of smart home. For instance, Steinberger and Michael (2018) propose a cognitive assistance architecture incorporating a "semantic" manual (using an ontology) to model instruction steps. However, these steps are neither hierarchical, nor is a planning system being used. Krieg-Brückner et al. (2015) propose an assistance system for planning and preparing meals. However, instead of using a standard planner (and their heuristics), cooking recipes are formalized in dynamic description logic (DDL) language employed by the SHIP tool. Georgievski et al. (2013) combine ontology modeling and HTN planning to recognize the activities of office users and to control the office appliances with the goal of saving energy. All of the above mentioned approaches focus mostly on the knowledge modeling aspect and do not employ dialog. An exception is the approach presented by Bercher et al. (2014), where planning and dialog are combined to assist users while setting up a home theater.

2 System Description

The developed prototype system serves to instruct novice users on how to accomplish DIY tasks with the help of power tools, such as electric drills, saws, sanders, etc. For this, step-by-step instructions are provided, where each step is illustrated by a text, a corresponding image and a video showing the activity to be performed (without a sound or voice track). Instructions are generated at two levels of granularity (steps and substeps). The interface offers the user to navigate forward and backward between steps to be performed consecutively, and to move from the more abstract level of instruction to the fine-grained level, and back. These options are presented schematically in Figure 1 (a). Figure 1 (b) shows one step in more detail. At the top, a progress bar gives an overview of the more abstract steps (which can be selected by clicking). Buttons can be used to request a video for the current step (videos are available at both levels), and clicking on highlighted concepts in the text yields a description of the corresponding concept (which is also output using text-to-speech, TTS). Besides touch and click gestures, speech and text input can be used to ask for information about the presented steps and the involved objects, e.g. what a "guide

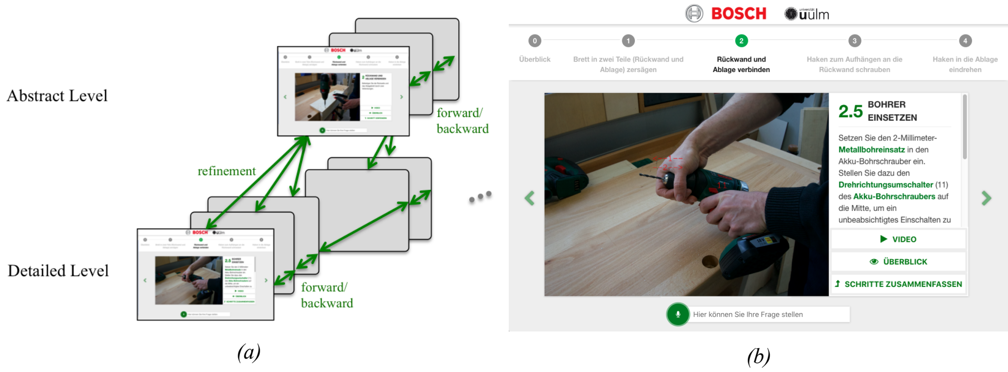

Figure 1: Navigational structure of instruction levels (a) and one detailed step (b) as shown by our prototype system

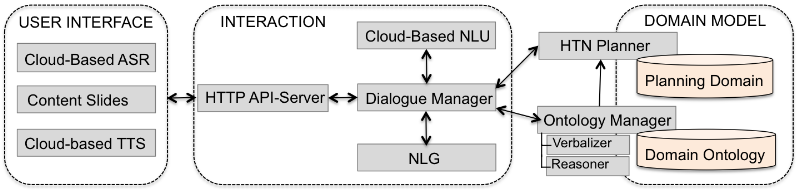

Figure 2: Companion system architecture of the prototype system

roller" (a part of the saw) looks like. A corresponding picture and text are then shown to the user, and the text is read out using TTS. Our tool is designed and tested using German language.

This functionality is based on an architecture that integrates hierarchical planning with an ontology-based knowledge representation and multi-modal dialog. Figure 2 shows the involved components. The domain model (on the right) incorporates formalized factual knowledge about tools and materials in an ontology, whereas procedural knowledge is formulated in the planning domain. Both formalisms are tightly interlinked, as described in (Schiller et al., 2017). This combined domain model allows for generating hierarchical plans and corresponding factual explanations (as simple natural language text) generated using a common source of information (the ontology). Thereby, the ontology management component employs both ontology reasoning (using HermiT, Glimm et al., 2014) und a translation mechanism from formal axioms to natural language text (so-called verbalization[1]). The interaction component (cf. Kraus et al., 2018) mediates between the user interface (on the left in Figure 2) and the knowledge representation components (on the right). User input is analyzed by an NLU component employing Microsoft's cloud-based *Language Understanding Intelligent Service* (LUIS) (Williams et al., 2015). The dialog manager implements the classical Information State Update approach (Larsson and Traum, 2000), which maintains the current dialog state. This state is updated by the dialog manager based on *dialog moves* (user input), requests sent to the planning- and ontology-based modules (yielding a plan or an explanation), and the system's own *dialog actions* (system output).

[1]https://verbalizer.github.io/

3 User Testing

To study the interaction with potential users, we devised a user study where participants have to perform a DIY task (constructing a wooden key rack) while interacting with the prototype assistant system. The system is provided on a tablet computer (with support for touch input). The presented task of building a key rack involves sawing one wooden plank into two parts of equal length using an electric jigsaw, connecting the parts with screws (using pre-drilling with an electric drill-driver), and attaching hangers and hooks. For testing purposes, participants are split into two groups, who are presented with two variants of the prototype:

(a) Full assistance: the system as described, providing instruction at two levels of granularity
(b) Baseline: only the lower level of granularity is shown. Questions are not supported.

This setup allows to observe differences in how the users interact with the system when experiencing a different range of system functionality. In the "full assistance" condition, participants are shown a short tutorial video on how to navigate the system (illustrating how touch and speech commands can be used with examples). Furthermore, we aim for qualitative insights into what problems, restrictions or difficulties are encountered in the two conditions. Participants' attitudes towards computerized assistance are assessed in a pre-test questionnaire and a post-test questionnaire, which also collects the participants' judgments regarding the usability of the presented system. In the following, we report our first observations from a recent user study with 18 DIY novices. Whereas the detailed analysis of the data is still underway, a number of aspects emerge from this study clearly enough to warrant further research and development. The pre-questionnaires document that the participants welcome to be helped by a digital assistant system, as a guide for using a new power tool (average score 4.3 on a five-point Likert scale, 1: disagree – 5: agree) and as a guide to completing a DIY project (average score 4.1). Whereas in the post-test-questionnaire, the different participants criticized different aspects (for some, speech input did not work as expected), they nearly unanimously indicated that they learned something about DIY while interacting with the application (average score 4.3). Similarly, they mostly judged the application to be useful (average 4.4). This is in line with our aim to foster users' knowledge and autonomy.

4 Challenges

Our approach differs from other forms of step-by-step guides by offering hierarchical levels of instructions that can be navigated by a user. Furthermore, it is capable of supporting different kinds of assistance requests using different input modalities. The first question concerns how the user can be introduced to the system's functionality in a straightforward way, and the second question concerns how this functionality is actually used, and what problems arise during use.

The first version of our prototype did not include the progress bar showing the abstract steps on top (see Figure 1). Users were able to navigate the hierarchical levels of instruction using buttons or speech, but first tests with this prototype showed that users stayed on the top level and rarely descended to the more detailed level. To more strongly prompt users to explore the functionality of the prototype, we changed the following:

- The wording of the button for jumping between hierarchical levels: initially the button for refinement was (ambiguously) labeled “mehr Info” (“more info”). Now it is labeled “Schritt verfeinern” (“refine step”).
- When the assistant starts the instruction, as the first step, the more detailed level is shown at first (instead of the first step at the abstract level as before).
- The video tutorial introducing the assistant system more prominently distinguishes between requests for changing the levels of instruction and other requests for information.

These measures make the hierarchical structure more explicit. However, the question through which path in the hierarchical structure the system should ideally take a user – in case the user does not explicitly request to be taken up or down – is still to be investigated further.

The second challenge concerns how to enable a user to tap the knowledge modeled (and offered) by the assistant. To efficiently use the available assistance, a user would profit from a clear understanding of what the system can offer and how to interact with it. Even though the tutorial video presented some commands and example questions (“what is ...”), users were not further instructed whether there are any restrictions that apply to the interaction (which come to bear in reality, if speech is not understood well, or if the knowledge model does not yield an adequate reply to the question). It turned out that users often avoided using speech in the study. On the one hand, this applies to simple commands (e.g. switching to the next step), where participants were more likely to take off the working gloves they were wearing to point/swipe on the tablet computer than to use a simple speech command instead. But this applies also to questions, where participants appeared to be reluctant as well. What could be the reasons for this?

First of all, users are likely to rely on a model of a system’s (or a dialog partner’s) capabilities and align accordingly (cf. Edlund et al., 2008). In our case, the system appears rather “quiet” – the images and videos did not use speech, only answers to questions about objects (e.g. “What is a PSR18Li2?”) were output using text-to-speech. Currently, the system is operated in responsive mode only where a user needs to trigger a request. Hence, it is not yet capable of initiating dialogues on its own, i.e. no pro-active support can be provided. We will investigate this in the next project phase since often users either do not know that support can be provided by the assistant, or they are not aware of how this information can be accessed.

We further observed that some users had practical difficulties with the execution of the most basic steps (e.g. using a screw clamp). These users took a long time and achieved suboptimal results (e.g. failed to accurately mark the wooden plank to be cut in half). This indicates that even the “detailed” level of instruction was not detailed enough for these users, and that some aspects of the DIY domain relevant for these users were not covered. A remaining challenge concerns how an assistant system could be enabled to detect that a user is struggling at the very basic level of execution.

5 Conclusion & Outlook

The prototype evaluation raised a number of design issues concerning the hierarchical organization of instructions, e.g. whether instruction for novice users should proceed top-down or bottom-up, and how to make the hierarchical structure more apparent to users. We further plan to extend the dialog capabilities to include more pro-activity. Future work will also include aspects of user modeling, such that the system can adapt the course of instruction to different users.

Acknowledgements

This work is funded by the German Research Foundation (DFG) within the technology transfer project "Do it yourself, but not alone: Companion Technology for Home Improvement" of the Transregional Collaborative Research Center SFB/TRR 62. The industrial project partner is the Corporate Sector *Research and Advance Engineering* of Robert Bosch GmbH. We thank Juliette Rose, Chunhui Zhu, Michael Dambier, and Mario Schmautz for their contributions to this work.

References

Behnke, G., Höller, D., & Biundo, S. (2018). Tracking branches in trees – A propositional encoding for solving partially-ordered HTN planning problems. In *Proc. of the First ICAPS Workshop on Hierarchical Planning* (pp. 40–47).

Behnke, G., Schiller, M., Kraus, M., Bercher, P., Schmautz, M., Dorna, M., … Biundo, S. (2018). Instructing novice users on how to use tools in DIY projects. In *Proc. of IJCAI-ECAI*, AAAI Press.

Bercher, P., Biundo, S., Geier, T., Hörnle, T., Nothdurft, F., Richter, F., & Schattenberg, B. (2014). Plan, repair, execute, explain – How planning helps to assemble your home theater. In *Proc. of ICAPS* (pp. 386–394). AAAI Press.

Edlund, J., Gustafson, J., Heldner, M., & Hjalmarsson, A. (2008). Towards human-like spoken dialogue systems. *Speech communication*, *50*(8-9), 630–645.

Georgievski, I., Nguyen, T. A., & Aiello, M. (2013). Combining activity recognition and AI planning for energy-saving offices. In *Proc. of UIC/ATC* (pp. 238–245). IEEE.

Glimm, B., Horrocks, I., Motik, B., Stoilos, G., & Wang, Z. (2014). HermiT: An OWL 2 reasoner. *Journal of Automated Reasoning*, *53*(3), 245–269.

Kraus, M., Behnke, G., Bercher, P., Schiller, M., Biundo, S., Glimm, B., & Minker, W. (2018). A multimodal dialogue framework for cloud-based companion systems. In *Proc. of the 10th International Workshop on Spoken Dialog Systems Technology (IWSDS 2018)*.

Krieg-Brückner, B., Autexier, S., Rink, M., & Nokam, S. G. (2015). Formal modelling for cooking assistance. In *Software, services, and systems* (pp. 355–376). Springer.

Larsson, S., & Traum, D. R. (2000). Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural language engineering*, *6*(3-4), 323–340.

Schiller, M., Behnke, G., Schmautz, M., Bercher, P., Kraus, M., Dorna, M., … Biundo, S. (2017). A paradigm for coupling procedural and conceptual knowledge in companion systems. In *International Conference on Companion Technology (ICCT 2017)*. IEEE.

Steinberger, C., & Michael, J. (2018). Towards cognitive assisted living 3.0. In *Proc. of the smarterAAL workshop co-located with PERCOM*, IEEE.

Williams, J. D., Kamal, E., Ashour, M., Amr, H., Miller, J., & Zweig, G. (2015). Fast and easy language understanding for dialog systems with microsoft language understanding intelligent service (LUIS). In *Proc. of SIGDIAL* (pp. 159–161). ACL.