Poster
ProNat: An Agent-based System Design for Programming in Spoken Natural Language

Sebastian Weigelt and Walter F. Tichy
Karlsruhe Institute of Technology (KIT)
Email: sebastian.weigelt@kit.edu, tichy@kit.edu

***Abstract*—The emergence of natural language interfaces has led to first attempts of programming in natural language. We present ProNat, a tool for script-like programming in spoken natural language (SNL). Its agent-based architecture unifies deep natural language understanding (NLU) with modular software design. ProNat focuses on the extraction of processing flows and control structures from spoken utterances. For evaluation we have begun to build a speech corpus. First experiments are conducted in the domain of domestic robotics, but ProNat's architecture makes domain acquisition easy. Test results with spoken utterances in ProNat seem promising, but much work has to be done to achieve deep NLU.**

I. Introduction

User interfaces have been growing in complexity for decades. While the command line was sufficient in the seventies, graphical user interfaces appeared in the eighties, web interfaces in the nineties, and touch screen interfaces in the last decade. The next generation of interfaces will handle unrestricted text and speech as input. Examples where talking to computers is reality today are navigation devices, Apple's Siri [1], Google's Voice Search [2], and several translation services. Thus, programming in natural language is within reach. It would enable anyone to program PCs, phones, or any programmable device. Therefore we propose programming with spoken natural language (SNL) to ease programming for layperson, a matter originally raised by Jane Sammet in 1966 [3], but no breakthrough was achieved since then. To promote programming with natural language we have designed ProNat. It unifies deep natural language understanding (NLU) with a novel software design. An agent-based architecture enables us to create agents for any NLU task, such as coreference resolution or named entity recognition (NER), independently. Thus, we can use both rule-based and probabilistic approaches. Furthermore our evaluation-driven development process allows quick integration and evaluation of agents. In contrast to existing approaches ProNat aims at interpreting longer and more complex spoken utterances. The system design also allows multi-modality (written texts, gestures). The domain is modeled in an ontology. First experiments are conducted in the domain of domestic robots, where we teach new skills to the household robot ARMAR-III [4]. The ARMAR-III robot has a broad set of basic skills such as moving around, grabbing things, and so on. New skills are composed of basic skills in combination with control structures (see figure 1 for an exemplary transformation).

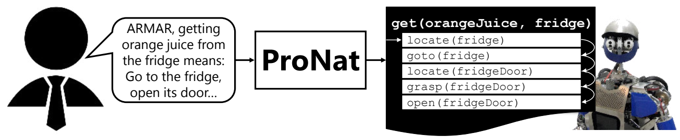

Fig. 1. Spoken user utterance to ARMAR-III state chart

II. Related Work

Since Sammet's proposal progress has been slow. A first influential contribution was NLC developed by Ballard and Biermann in 1979 [5]. NLC allowed matrix calculation in written natural language. Even though its vocabulary was restrict to the domain and functionality remained limited, NLC showed that programming in natural language is possible. Other approaches use fixed domains, but made progress concerning expressiveness and functionality [6][7]. Liu and Lieberman claim that the expressiveness of natural language (English) is sufficient for programming. Their prototype Metafor creates classes and method stubs from user stories, but leaves the implementation to human developers [8]. Even though Metafor produces only stubs, it indicates that natural language could indeed be used as a programming language. In prior work we demonstrated generation of UML diagrams from textual specifications [9] and script-like programming in written natural language [10]. Intelligent assistants such as Siri, GoogleNow or Cortana can deal with spoken utterances but answer questions or interpret single commands [1][2].

III. Design Principles

Agent-based: An agent-based design allows continuous integration of new functionality. Internal processing of an agent is independent and transparent to other agents and may be probabilistic or rule-based. Furthermore, agent performance can be evaluated individually. A shared data-storage serves as interface between the agents. All other system elements, such as input processing (automatic speech recognition, gesture recognition) and ontology connection, are integrated as modules.

Evaluation-driven: An evaluation-driven development implies continuous evaluation of the system or parts of it to ensure progress. The effectiveness of this approach can be enhanced by using realistic examples. We therefore started to collect speech utterances. The utterances that build our speech corpus comprise complex instructions for a household robot. With test sets from the corpus we are able to benchmark continuously. Moreover, we may encounter new challenges, opportunities and problems by working with realistic examples.

Knowledge-based: The use of knowledge resources makes deeper NLU possible, especially regarding disambiguation. ProNat offers connections to world-knowledge databases (KDB) such as Cyc [11] and WordNet [12]. The target domain is modeled as an ontology, which enables the use of domain knowledge for NLU.

Domain independent NLU: ProNat has the ambition to make most of the NLU processing domain independent. The advantage is that new domains can be acquired more easily. Domain knowledge is only accessed through its well-defined ontology representation. If the domain changes, only the content of the ontology changes, while the structure remains the same. This behavior leaves domain dependent NLU agents unaffected.

IV. Architecture

The architecture of ProNat, as shown in figure 2 is centered around a graph-based data storage. The graph acts as a) representation of the spoken utterance and b) shared data storage for NLU agents. Results produced by agents are made visible to other agents by graph transformations. Input recognition is encapsulated in modules. Thus various off-the-shelf automatic speech recognizers (ASR) can be integrated easily. Speech input is processed in a lightweight shallow natural language processing (NLP) module. It builds an initial graph from part-of-speech, chunking, and semantic role labeling information. If the ASR offers alternative transcriptions, multiple graphs are created and processed simultaneously. The domain ontologies are connected via a shared interface. Domains specific features are added through the usage of adapters. Note that only a new adapter has to be implemented for each acquired domain. The ontology connection module serves as interface for agents requesting domain knowledge and also as interface for code generation.

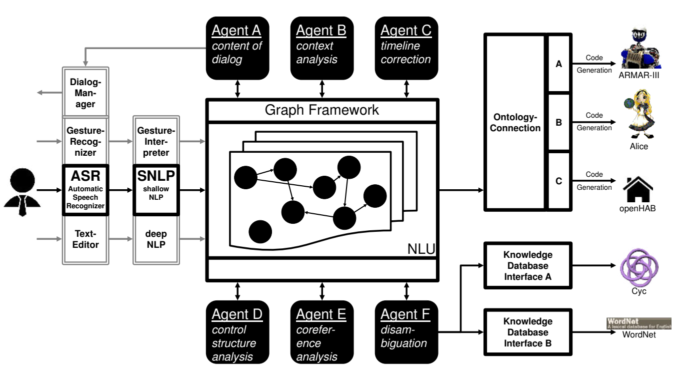

Fig. 2. Architecture of ProNat

V. Conclusion and Prospect

We propose an agent-based system design for deep NLU. It offers a combination of probabilistic and rule-based methods. We concentrate on spoken input in the domain of domestic robotics, but both input and target domain may be replaced. The goal is to extract processing flows and control structures from spoken utterances that can be mapped to programming constructs (classes, methods, loops etc.). World and domain knowledge are integrated for disambiguation. An evaluation-driven development with realistic benchmarks ensures progress. First experiments show that the approach works well if the ASR word error rates are low. Domain dependent training data is needed for the time being. Although we have begun to build a speech corpus we must deal with word errors first. We hope to compensate such flaws through intensive context analysis and disambiguation with the help of domain and world knowledge. We can do so since we consider multi-sentence utterances.

References

[1] J. R. Bellegarda, "Spoken language understanding for natural interaction: The Siri experience," in *Natural Interaction with Robots, Knowbots and Smartphones*, J. Mariani, S. Rosset, M. Garnier-Rizet, and L. Devillers, Eds. Springer New York, 2014, pp. 3–14.

[2] C. L. Ortiz, "The road to natural conversational speech interfaces," *Internet Computing, IEEE*, vol. 18, no. 2, pp. 74–78, Mar 2014.

[3] J. E. Sammet, "The use of English as a programming language," *Commun. ACM*, vol. 9, no. 3, pp. 228–230, Mar. 1966. [Online]. Available: http://doi.acm.org/10.1145/365230.365274

[4] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*. IEEE, 2006, pp. 169–175.

[5] B. W. Ballard and A. W. Biermann, "Programming in natural language: NLC as a prototype," in *Proceedings of the 1979 annual conference*, ser. ACM '79. New York, NY, USA: ACM, 1979, pp. 228–237.

[6] V. Le, S. Gulwani, and Z. Su, "Smartsynth: Synthesizing smartphone automation scripts from natural language," in *MobSys'13*, vol. 2, no. 3, 2013, p. 5.

[7] R. Knöll and M. Mezini, "Pegasus: first steps toward a naturalistic programming language," in *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, ser. OOPSLA '06. New York, NY, USA: ACM, 2006, pp. 542–559.

[8] H. Liu and H. Lieberman, "Metafor: Visualizing Stories as Code," in *IUI '05: Proceedings of the 10th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2005, pp. 305–307. [Online]. Available: http://portal.acm.org/citation.cfm?id=1040830.1040908

[9] M. Landhäußer, S. Körner, and W. Tichy, "From requirements to uml models and back: how automatic processing of text can support requirements engineering," *Software Quality Journal*, vol. 22, no. 1, pp. 121–149, 2014. [Online]. Available: http://dx.doi.org/10.1007/s11219-013-9210-6

[10] M. Landhäußer, T. Hey, and W. F. Tichy, "Deriving timelines from texts," in *Proceedings of the 3rd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering*, Jun. 2014, pp. 45–51.

[11] Cycorp Inc., *ResearchCyc*, 31.07.2014. [Online]. Available: http://www.cyc.com/platform/researchcyc

[12] G. A. Miller, "Wordnet: a lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748