Controlled Natural Languages for Interface Agents (Extended Abstract)

Tamás Mészáros
Budapest University of Technology and Economics
Magyar tudósok krt. 2. Bldg. I. Room E437
Budapest, Hungary, H-1117
+36 1 463-2899
meszaros@mit.bme.hu

Tadeusz Dobrowiecki
Budapest University of Technology and Economics
Magyar tudósok krt. 2. Bldg. I. Room E437
Budapest, Hungary, H-1117
+36 1 463-2899
dobrowiecki@mit.bme.hu

ABSTRACT
Interface agents help users to deal with the complexity of today's computer systems. It is commonly required that these agents should employ natural language to communicate with humans. Natural language processing is, however, a complex and difficult task for agent designers. Authors propose a new framework that significantly reduces the complexity of building such interfaces. It relies on the concepts of application-specific controlled natural languages, a guided user interface, and a conceptual model of the interface functions that allows the automatic generation of controlled language.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: *Natural language user interfaces.*

I.2.7 [**Natural Language Processing**]: *Language parsing and generation.*

I.2.11 [**Distributed Artificial Intelligence**]: *Intelligent agents.*

General Terms
Algorithms, Design, Languages, Theory.

Keywords
Controlled natural language, interface agent, conceptual graph, natural language processing, human-agent communication, model-based language generation.

1. INTRODUCTION
In order to be more natural and comprehensible to the human user an ideal interface agent should converse with humans in a free-form natural language just like humans do.

The task of general natural language understanding, however, is still very hard to solve in today's software systems. Several techniques were proposed to overcome this problem. Restrictions on the user interface (like using template- or menu-based communication [8]) could significantly ease the implementation of the communication but they constrain the expressiveness and applicability much. Building free-form natural language communication [3] with domain restrictions is also possible but it requires the application of very complex language tools (parsing, disambiguation, understanding user's intentions) and detailed knowledge base that are not easy to create and maintain.

We propose to design an interface agent around a controlled natural language, which provides a good trade-off between interface restrictions and the complexity of general natural language understanding. A new interface technique based on automatically generated controlled natural languages is also proposed.

Cite as: Controlled Natural Languages for Interface Agents (Extended Abstract), Tamás Mészáros, Tadeusz Dobrowiecki, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra, and Castelfranchi (eds.), May, 10–15., 2009, Budapest, Hungary, pp. XXX-XXX.

2. CONTROLLED NATURAL LANGUAGE
Controlled natural language [1] resembles the ordinary languages but it has a strict (and restricted) set of language rules, vocabulary and unambiguous meaning.. These restrictions allow the successful processing of a controlled natural language by avoiding disambiguation and uncertain grammar rules, and by explicitly linking the language to the contextual knowledge.

Unfortunately, several problems with controlled languages prevented their widespread application. Firstly, the controlled language is not exactly the same as the natural language known to the user – the user has to learn it. This could take time, but often there is no time to train the user (or there is little interest in learning the restricted rules and vocabulary). The user might “adapt” to the language with time, but it is usually required (or desired) that the user should be able to use it from the very beginning.

Secondly, these languages are not flexible enough, they could not adapt to the changes in the underlying software system (e.g. in data structures). These changes usually render the controlled language obsolete – it has to be adapted manually. Such task requires programmers with a unique training, and the adaptations are usually not easy to perform.

To overcome these limitations we propose restrictions in general natural language interfaces and model-based language generation to ease the authoring and maintenance of controlled languages.

3. RESTRICTIONS TO NATURAL LANGUAGE AGENT INTERFACES
In order to coerce the user to follow the rules of the controlled language we propose restrictions in the user interface. These restrictions prevent the user to diverge from the rules and

vocabulary of the controlled language by monitoring the user's input and automatically adapting it to the language. They also provide help to the user on how to use the restricted language by suggesting possible language constructs, words, or expressions.

A simple way to implement restrictions is the predictive text input [6]. This method constantly analyzes the text the user types in, it determines the set of possible sentences based on that input, and provides suggestions to continue the typing at the cursor position.

With this solution we attain two goals. The text input is constructed following the rules of the language and the user gets immediate help in using the controlled language. The user can use the interface from the very beginning (it might be slow for the first time, yet it will be usable), and with time the interface language will become increasingly familiar and easy to use.

This solution raises additional requirements for natural language processing. It should support processing sentence fragments and the generation of "continuations" (feature extensions). Furthermore, language processing and generation techniques applied in the interface should be efficient. They have to analyze the user input and provide suggestions in the real time while the user is typing. The details of effective language representation and how restricted languages are used in controlled user interfaces are described in more detail in our Web page [4].

4. AUTOMATIC CONSTRUCTION

To facilitate the design process we developed a model-based language generator that is similar to model-based software development in a sense that they are both based on a domain model that is used (along with a set of pre-made application components) to create the full application (the controlled language interface in our case).

Our goal was to provide a modeling framework that makes the design of controlled natural language agent interfaces relatively easy. Programmers without any knowledge of natural language processing techniques should be able to create and maintain controlled language interfaces. To this aim language grammars and vocabularies should be automatically generated from the model of the user interface itself.

After evaluating several modeling techniques we have selected the Conceptual Graph (CG) modeling technique [5]. It is a very flexible tool to define concepts and their relations. It allows the creation of loosely defined concept models, but it also makes it possible to construct detailed models supporting logic-based reasoning. These graphs have been already applied in similar applications [7], and they were also proposed as a natural language grammar representation called Conceptual Graph Grammar [2].

CGs are capable of representing the interface concepts and relations in an easy to understand way but they lack some features required to generate the desired controlled language interfaces. Therefore, we have extended the CGs with methods to describe two further levels of the same model: the data and the language. These modifications do not change the basic behavior of the CGs but extend the applicability of the model.

The new data level introduces bindings to the application. At this level concepts (and concept types) defined in the CG model can specify relations to the particular data sources in the application. Application bindings make it possible to dynamically create concept instances during the operation of the NL interface.

The language level adds special attributes to concepts and relations. The CG model itself is language-independent. Concepts and relations could be named using any notation or language. Our language extensions introduce language-specific constructs and symbols. They specify how a given relation or concept is represented in a given language. Language attributes of relations identify grammar constructs, while concept attributes define symbols in the context of languages and relations.

From the model we generate the controlled language in two steps: first relations (and concepts) form the basis of the language, the grammar, then the vocabulary is constructed from concepts, their language extensions and data bindings.

5. CONCLUSION AND FURTHER WORK

We proposed an approach to effectively utilize controlled natural languages in human-agent interfaces. The approach is based on the idea of using an automatically generated application-specific controlled natural languages in a restricted user interface.

The proposed approach has two principal advantages. First, it allows human-agent communication in (controlled) natural languages. Secondly, it facilitates the automatic generation and maintenance of the applied controlled language which eases the implementation of such interfaces even for programmers without deep knowledge in natural language processing. These advantages contribute to the successful application of natural languages in human-agent interfaces.

Other details of our research and online demonstrations available at

http://www.mit.bme.hu/~meszaros/work/cnl/

6. REFERENCES

[1] Allen, J., Barthle, K., 2004. Introductory overview of Controlled Languages. Society for Technical Communication meeting of the Paris, France chapter. 2 April 2004.

[2] Johnson, S.B., 1998. Conceptual Graph Grammar – A Simple Formalism for Sublanguage, Methods Inf Med. 37 (4-5): pp. 345-52.

[3] Kemke, C., 2006. Natural Language Communication between Human and Artificial Agents, Artificial Agents, Pacific Rim International Workshop on Multi-Agent Systems PRIMA-2006, Guilin, China. pp. 84-93

[4] Mészáros, T., Controlled Natural Language interfaces, http://www.mit.bme.hu/~meszaros/work/cnl/

[5] Polovina, S., 2000. Conceptual Graphs: An Overview, Software and Issues, A nearplanet.com White Paper

[6] Smith, S.L., Goodwin, C.N., 1971. Alphabetic Data Entry Via the Touch-Tone Pad: A Comment, The Mitre Corporation, HUMAN FACTORS, 13(2) pp 189-190.

[7] Sowa, J.F., 1976. Conceptual Graphs for a Data Base Interface, IBM Journal of Research and Development 20(4), pp. 336–357

[8] Thompson, C., Pazandak, P., Tennant, H. 2005. Talk to your semantic Web, Internet Computing, IEEE, vol. 9, pp. 75-78.