# A Spoken Dialogue System to Control Robots

**Hossein Motallebipour, August Bering**

Dept. of Computer Science, Lund Institute of Technology,
SE-221 00 Lund, Sweden;
E-mail: d97hm@efd.lth.se, d98abe@efd.lth.se

## Abstract

Speech recognition is available on ordinary personal computers and is starting to appear in standard software applications. A known problem with speech interfaces is their integration into current graphical user interfaces. This paper reports on a prototype developed for studying integration of speech dialogue into graphical interfaces aimed towards programming of industrial robot arms. The aim of the prototype is to develop a speech dialogue system for solving simple relocation tasks in a robot workcell using an industrial robot arm.

## 1 Introduction

Industrial robot programming interfaces provide a challenging experimental context for researching integration issues on speech and graphical interfaces. Most programming issues are inherently abstract and therefore difficult to visualize and discuss, but robot programming revolves around the task of making a robot move in a desired manner. It is easy to visualize and discuss task accomplishments in terms of robot movements. At the same time robot programming is quite complex, requiring large feature-rich user interfaces to design a program, implying a high learning threshold and specialist competence. This is the kind of interface that would probably benefit the most from a multi-modal approach.

This paper reports on two extensions to an earlier prototype speech user interface developed for studying multi-modal user interfaces in the context of industrial robot programming (0). The extended prototype gives the robot the ability of understanding spoken natural language instructions and perform simple tasks. The user/operator will be able to refer to objects in the robot's environment either spatially, or using descriptive object names. The prototype is restricted to manipulator-oriented robot programming. Examples of spoken instructions that the robot should be able to understand and perform are:

> Robot, please move 10 steps to the right.
> Move up slightly.
> Grip the cube.
> Move forward and a bit up.
> Please move 15 steps to the right and down to the table.

The spoken language instructions are used within a restricted task domain. This has several advantages:

- The speech vocabulary can be quite limited because the interface is concerned with a specific task. The number of natural sentences tend to be limited as well.
- A complete system decoupled from existing programming tools may be developed

to allow precise experiment control.

- It is feasible to integrate the system into an existing tool in order to test it in a live environment. The prototype could be integrated into existing CAD software where it would enhance a dialogue, or a design tool, in the larger CAD tool.

Further motivation for keeping speech vocabularies limited lies in the fact that current available speech interfaces seem to be capable of handling small vocabularies efficiently, with performance gradually decreasing as the size of the vocabulary increases. This also makes it interesting to examine the impact of small domain-specific speech interfaces on larger user interface designs, perhaps having several different domains and collecting them in user interface dialogues.

The general purpose of the prototype is to provide an experimental platform for investigating the usefulness of speech in robot programming tools. The high learning threshold and complexity of available programming tools makes it important to find means to increase usability. The prototype extensions presented in this paper are summarized below:

- Implemention of a human-robot dialogue system that is capable of handling spatial references and named references to workspace objects.

- Utilization of XML for experimental setups in order to test different dialogue situations. This includes modifying experiment geometry (robots and workspace) as well as using different speech grammars and vocabularies.

Organization of this paper is as follows: we will first take a look at the methods used for the implementation, such as ASR and NLP. Then, the experiment and the prototype itself are presented. A subjective evaluation and results of the implementation and the experiments with the prototype are then presented. The paper will conclude with a short discussion about the result.

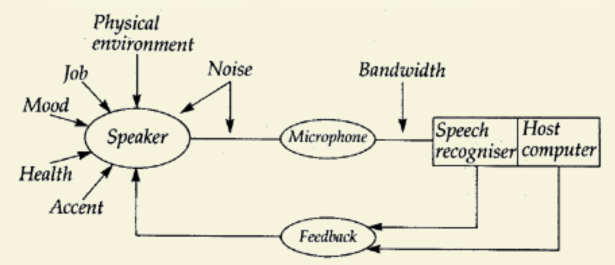


Figure 1: Components of a speech recognition system and factors affecting system performance. See (0).

## 2 ASR, NLP and CFG

An overview of dialogue systems is given in (0). The language tools used in this paper are basically automatic speech recognition (ASR) combined with natural language processing (NLP) using context-free grammars (CFG).

### 2.1 Automatic Speech Recognition (ASR)

ASR could be defined as the ability of machines to recognize human speech in a specific language.

There are three basic uses of ASR:

- Command and control: give commands to the system that it will then execute. Systems for this purpose are usually speaker-independent.

- Dictation: spoken sentences will be transcribed into written text. Systems for this purpose are usually speaker-dependent.

- Speaker verification: the voice is used to identify a person uniquely.

The common components of an ASR system include the person speaking to the system, input devices to the system (i.e. microphones) and the ASR system itself.

An ASR system is shown in Figure 1. The figure show factors affecting the performance of an ASR system, for example health and mood of the speaker.

## 2.2 Natural Language Processing (NLP)

NLP is about building computational models for understanding natural language. NLP models will, from a natural language text, compute a representation of the semantic meaning of that text.

Several levels of analysis and knowledge are commonly applied in NLP (0):

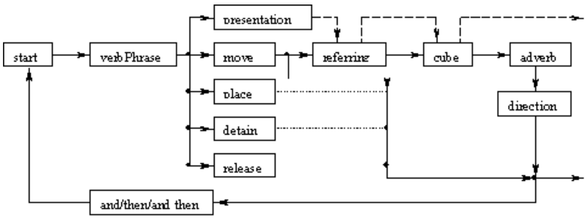
- Morphological analysis looking into the construction of words, prefixes and suffixes.
- Syntactical analysis using the structural relationships between words.
- Semantical analysis finding the meanings of words, phrases, and expressions.
- Discourse analysis to find the relationships across different sentences or thoughts with contextual effects taken into account.
- Pragmatic analysis looking for the purpose of a statement trying to investigate what the used language is used to communicate.
- Applying world knowledge (facts about the world at large, common sense) for interpreting sentences in a general context.

NLP is attractive and has several application areas like database query interfaces, machine translation, fact extraction, information retrieval / search engines, categorization, language filtering, text summarization, question answering systems, speech recognition and spoken language understanding and intelligent tutoring systems.

## 2.3 Context Free Grammars (CFG)

Many grammars used for NLP systems are CFG since they have been widely studied and understood and hence highly efficient parsing mechanisms have been developed using them.

In basic terms, a CFG define sentences that are valid using a parse tree. The parse tree breaks down the sentence into structured parts that can be easily understood and processed. A parse tree is usually constructed using a set of rewrite rules which describes legal language structures.



Dotted lines replace different combinations with referring, cube, adverb and direction. Broken line represents the output from the function \emph{presentation}.

Figure 2: Scheme showing outline of implementation of prototype CFG grammar.

In the definition of the *grammar rules* a *state graph* can be used to illustrate how sentences are to be constructed. Each sentence following the paths in the graph will be recognized as a correct phrase. For instance, the semantic meaning: *Grip cube number 1!* should accept phrases like:

Robot, please grip cube number 1
Robot, please grab cube number 1
Robot, please grasp cube number 1

and:

Robot please grip the cube number 1
Please grip the cube number 1
Robot grip the cube number 1
Grip the cube number 1
Grab the number 1
Grab cube 1
...

The scheme in the figure 2 show an outline of the graph of the CFG grammar for controlling the robot arm in the prototype. Two paths in the grammar are marked. The straight line at the bottom pointing to the right corresponds to the sentence: *Robot, please move the cube number one slightly to the right.* The broken line at the top of the scheme corresponds to the sentence: *This is cube number 2.*

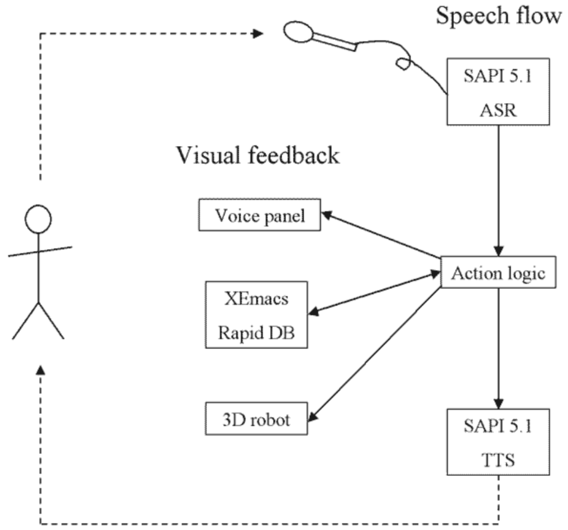


Figure 3: Prototype system dataflow.

## 3 The Prototype

The prototype presented here is a user interface where speech has been chosen to be the primary interaction modality but is used in the presence of several feedback modalities. Available feedback modalities are text, speech synthesis and 3D graphics.

The prototype system utilizes the speech recognition available in the Microsoft Speech API 5.1 software development kit (SAPI). SAPI can work in two modes: command mode recognizing limited vocabularies and dictation mode recognizing a large set of words using statistical word phrase corrections. The prototype uses the command mode. It is thus able to recognize isolated words or short phrases.

The system architecture (see Figure 3) consists of several applications:

- The *ASR application* uses SAPI 5.1 to recognize a limited domain of spoken user commands. Visual feedback is provided in the Voice Panel window. Recognized words and phrases are received from the SAPI 5 ASR engine graded with a confidence value. This information, as well as extracted semantic information, is sent to the action logic application.
- The *Action Logic application* controls the user interface system dataflow and is the heart of the prototype. Basically it receives phrases from the ASR application and acts upon them. For instance, if the semantic information of a phrase includes robot arm movement, corresponding RAPID code is generated for the robot[1]. A phrase that reads *Move two steps left*, will generate the RAPID code `MoveL (0,2,0)`. In this instance the RAPID code will be sent to the 3D robot application for execution providing 3D feedback, and to the XEmacs application for storage and textual feedback.
- The *Text-To-Speech application* provides user voice feedback.
- The *XEmacs application* acts as a database of robot movement commands written in the robot programming language RAPID, since it is an editor it also allows direct editing of RAPID programs.
- The *3D Robot application* provides a 3D visualization of the robot arm with workspace. It understands and can perform a subset of RAPID commands.

The applications forms a distributed system. Inter-application communication is performed using TCP/IP.

The ASR application uses SAPI 5 in command mode (as opposed to the also available dictation mode). The command mode uses CFG grammars to recognize single words and short phrases. The CFG format in SAPI 5 defines the structure of grammars and grammar rules using XML[2]. In the prototype, this XML format is used for implementing the prototype NLP capabilities.

### 3.1 SAPI 5 XML CFG Grammar Format

The reference document describing the XML SAPI 5.0 speech recognition grammar

[1] RAPID is a programming language for industrial robot arms developed and used by the ABB company.

[2] A SAPI 5 included XML CFG grammar application compiles CFG XML grammars into the binary format required by the SAPI 5 speech recognition engine.

format (based on the Microsoft schema language and not fully W3C compliant[3,4]) is included in the SAPI SDK documentation.

Below is an example of a grammar rule written in SAPI 5 XML:

```
<RULE NAME="grip">
   <LIST>
        <P>grip</P>
        <P>grab</P>
        <P>grasp</P>
   </LIST>

   <P>cube</P>

   <LIST PROPID="CUBENR">
        <P VAL="1">one</P>
        <P VAL="2">two</P>
        <P VAL="3">three</P>
   </LIST>
</RULE>
```

The grammar rule corresponds to sentences like "grip cube two". Only words between <P> tags are recognized. Furthermore, the rule is augmented with semantic information (enclosed as name-value pairs within XML tags). This information is extracted during sentence recognition by the ASR application and provides the means for a simple context-independent NLP analysis performed by the prototype. The sentence "grip cube two" would provide the ASR application with the following semantic information:

RULE: grip
CUBENR: 2

## 4 Experiment

A series of Wizard-of-Oz experiment transcripts were recorded before the work on the prototype began. Below is an example of a dialogue between the user and the system derived from the transcribed Wizard-of-Oz experiments:

Robot, please move 10 steps to the right!
Move down to the table!
Move up slightly!
Move 1 step to the right! Move down!
Grip!
This is cube 1.
Move forward and a bit up!
Move 4 steps to the left!
Move a bit down and drop the cube!
Move up slightly!
Could you move 15 steps to the right
and down to the table?
Grab the cube!
This is cube 2.
Put it on cube 1!
Please move 2 steps up and 6 steps left!
Move 2 steps down!
Grab the cube number 3!
Put it on the cube number 2!

The robot knows the position of the table. It has no information about the cubes and where they are situated. The user should guide the robot arm to each cube, where each cube is denoted a specific name by the user, e.g. cube number one. The robot remembers the location of the specified cubes.

The goal for the user is to instruct a virtual robot arm, using natural spoken English, to identify and put three cubes on top of each other on the table. Figure 4 shows the experimental setup as well as the user interface.

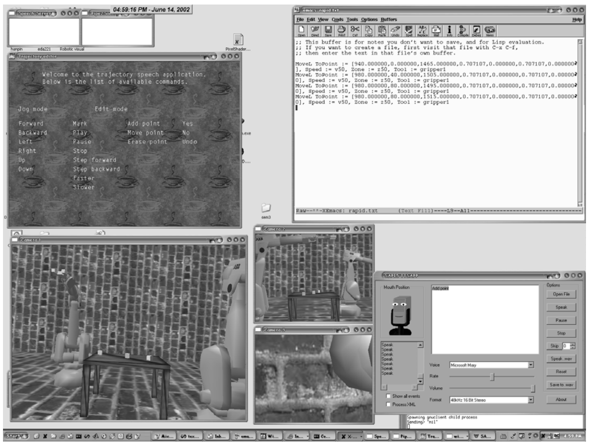
Three non-native English-speakers has tested the ASR and NLP part of the prototype system using dialogues similar to the one above. Dialogue sentences are recognized with good accuracy using SAPI 5. However, at the time of the test the prototype implementation partially lacked 3D and textual feedback for part of the dialogue. Evaluation of the prototype system with full multimodal feedback will be performed at completion of these parts.

## 5 Discussion

The experiment with three subjects showed the SAPI command mode and the CFG grammar used in the presented prototype to be

[3] The World Wide Web Consortium (W3C), http://www.w3.org

[4] Although the MS Speech SDK (SAPI 5.1) documentation says that the schema will be rewritten and compliant with W3C once it has been approved by W3C.


04:59:16 PM - June 14, 2002
Speech Server
Trajectory editor
Camera 1
emacs: rapid.txt
File Edit View Cmds Tools Options Buffers Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News

```
Welcome to the trajectory speech application.
Below is the list of available commands.

Jog mode              Edit mode

Forward       Mark            Add point     Yes
Backward      Play            Move point    No
Left          Pause           Erase point   Undo
Right         Stop
Up            Step forward
Down          Step backward
              Faster
              Slower
```

```
;; This buffer is for notes you don't want to save, and for Lisp evaluation.
;; If you want to create a file, first visit that file with C-x C-f,
;; then enter the text in that file's own buffer.

MoveL ToPoint := [940.000000,0.000000,1465.000000,0.707107,0.000000,0.707107,0.000000
], Speed := v50, Zone := z50, Tool := gripper1
MoveL ToPoint := [980.000000,40.000000,1505.000000,0.707107,0.000000,0.707107,0.00000
0], Speed := v50, Zone := z50, Tool := gripper1
MoveL ToPoint := [980.000000,80.000000,1495.000000,0.707107,0.000000,0.707107,0.00000
0], Speed := v50, Zone := z50, Tool := gripper1
MoveL ToPoint := [980.000000,80.000000,1515.000000,0.707107,0.000000,0.707107,0.00000
0], Speed := v50, Zone := z50, Tool := gripper1

Raw--**-XEmacs: rapid.txt      (Text Fill)----L9--All---------------------------------
```

Figure 4: The prototype system user interface consists of four windows; 1. The voice panel containing lists of available voice commands. 2. The XEmacs editor containing the RAPID program statements. 3. The 3D visualization showing the current state of the hardware. 4. The TTS application showing the spoken text.

rather stable. The feedback from the system gave clear signals that it could hear and transcribe the spoken sentences well. Subjective impressions from test subjects were positive.

The dictation mode of SAPI 5 were tried in the initial stages of prototype development. The mode uses a large set of words and should potentially suppart a larger set of English sentences than the chosen solution. However, recognition accuracy proved insufficient. The command mode with smaller vocabulary was more accurate. Although the grammar and set of used words in the system is limited the test subjects felt the dialogue came natural.

## 6 Conclusion

A prototype user interface for examining spoken dialogues for controlling simple relocation tasks to be performed by robot arms has been developed.

- The prototype uses SAPI 5 and CFGs for processing and understanding spoken natural language robot instructions.

- The prototype dialogue system supports spatial referencing with respect to the robot arm and identification and object referencing by name in the robot workspace.

- Feedback is provided by several modalities.

## 7 Acknowledgements

We would like to thank Pierre Nugues and Mathias Haage at the Department of Computer Science at Lund University for valuable insights and comments during our work.

## References

M Haage, S Schötz and P Nugues. *A Prototype Robot Speech Interface with Multimodal Feedback*. Proceedings of the 2002 IEEE, Int. Workshop on Robot and Human Interactive Communication. Berling, Germany, Sept. 25-27, 2002.

J Hollingum and G Cassford. *Speech Technology at Work*. New York: IFS Publications Ltd, 1988

*Microsoft SAPI homepage*. http://research .microsoft.com/srg/sapi.aspx. 2003.

P Nugues. *Lecture Notes: Introduction to Language Processing and Computational Linguistics*. 2002. Contact P Nugues at Department of Computer Science, Lund Institute of Technology, Sweden.

*Spoken dialogue technology: enabling the conversational user interface*, ACM Computing Surveys (CSUR), vol. 34, nbr 1, 2002, pp. 90-169, ACM Press.