Corpus–Based Test System for an Automated Speech Processing System *

L.J.M. Rothkrantz, R.J. van Vark, and H. Koppelaar

Knowledge Based Systems
Applied Computer Science
Delft University of Technology
email: alparon@kgs.twi.tudelft.nl

Abstract

At this moment Automated Speech Processing is applied in the design of automated telephonic inquiry systems to simulate a human operator. To test successive releases of those systems an automated test system is developed in which the caller is simulated. The Simcall system is based on human–human dialogues. The system will be described as well as some preliminary test results.

1 Introduction

The Dutch company OVR[1] provides information by telephone, concerning public transport services in the Netherlands. Last year, the company handled over 12 million requests for information. Recently, an Automated Speech Processing (ASP) prototype has been developed for OVR to reduce the load on their call centers. This prototype is based on the prototype developed for the German railways [Aust et al., 1995, Oerder and Aust, 1994]. As the humanly operated system is efficient, effective and highly appreciated by callers, a corpus–based approach was chosen. However, imitating a human operator does not guarantee a successful system, as human–human dialogues differ from human–machine dialogues in many aspects. Using a corpus–based approach, the designers expect to transfer the merits of human–operated systems to the ASP system.

The ASP prototype was evaluated in a survey study. Over 500 respondents took part in this survey study by calling the system and filling out a questionnaire. The analysis of this survey study is presented in [Rothkrantz et al., 1997]. This way of evaluating an ASP system is very time consuming. Another problem is that respondents are seldom interested in the result of the information query, as they will probably never make the journey they requested information about.

As developing an ASP system is not easily realised, the developed prototype needs further improvement and additional training and testing. The ultimate goal of OVR is to have an ASP system that is an automated version of the humanly operated system. To

*This work is funded by OVR and Senter.

[1] OVR is an acronym for the Dutch translation of Public Transport Travel information.

obtain an objective assessment of the progress made in every new release, a corpus–based testbed is called for. If an ASP system handles a representative sample of human–human dialogues adequately, the system is considered to be equivalent to the human operator.

The designers of an ASP system have to simulate a human operator, while we have to simulate a caller to develop our testbed, Simcall. In general, human dialogues have no fixed scenario, but are very flexible and unpredictable. In the OVR corpus, most dialogues follow an underlying scenario. However, a great variation of linguistic expressions can be used within this general framework. An ASP system has to handle all these different expressions. Therefore, our Simcall system contains a representative sample from the set of linguistic expressions found in the OVR corpus.

As stated before human–human and human–machine dialogues are quite different. As long as speaker independent speech recognition is far from perfect, ASP systems have to verify most of the expressions they recognise (or think they have recognised). As a consequence, designers of an ASP system have to spent a lot of effort in developing error correcting procedures. In Simcall, we do not test the success rate of error correcting procedures, because these procedures cannot be corpus–based as verification is not often found in human–human dialogues. This is one of the most essential differences between human–human and human–machine dialogues. As an ASP system cannot behave as a human operator at this moment, we restrict ourselves to the following questions, of which some will be answered using future versions of Simcall:

- How many of the linguistic expressions from the corpus of human–human dialogues are correctly recognised and processed by the ASP system?
- Is the ASP system speaker dependent?
- Is the ASP independent of background noise?

In the next sections we will describe the Simcall system as well as some preliminary test results.

2 OVR dialogue model

Simcall is developed to test VIOS [Veldhuijzen van Zanten, 1996], the ASP prototype of OVR. In VIOS, a request for information on travelling by public transport consists of 5 topics: departure place, arrival place, arrival time, departure time, and date of travelling. A human operator or an ASP system is focussed on the filling of these five information slots.

VIOS uses a so–called "blackboard" approach, in which the client is requested to provide the information to fill each slot. The ASP system extracts the relevant information from the client's utterances which can occur in any order. In the current application only exact arrival and departure times can be requested by the client. If all appropriate slots are filled, the ASP system can query a database to retrieve a travel schedule and provide it to the client.

Although it is possible for the client to provide all information in one sentence, the information is usually provided step-by-step. After every utterance of the client, the ASP system asks for verification, clarification or new information. Because the available information system is restricted in place and time, it acts as a finite state machine. The

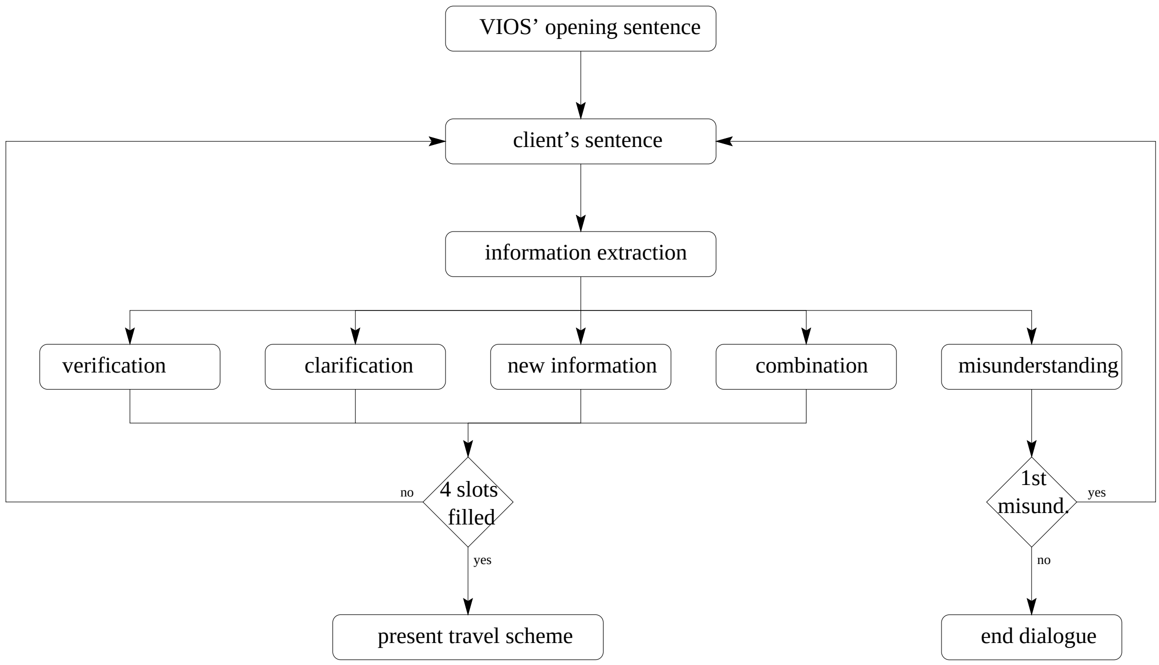

Figure 1: Data flow in a VIOS dialogue

dialogue manager of the ASP system can generate a finite number of prompts, which can be ordered as follows (see figure 1):

Additional information
The ASP system requests further information if two or more slots are open. The last open slot will be the arrival or departure time, which is requested by the caller. These prompts are typically given as wh-questions: "Which railway station do you want to travel to?" and "What time do you want to travel?"

Verification
The system always asks for verification of the information extracted from the client's utterances. For example, "You want to travel on [date] from [departure place] to [arrival place]?"

Clarification
The ASP system reduces ambiguity by requesting additional information or verifying the systems interpretation. For example, "Do you want to depart at 8 o'clock a.m. or 8 o'clock p.m.?"

Correction
The system requests correction of wrongly extracted information. For example: "You don't want to travel from Delft, please would you be so kind to make a correction."

Combination
The ASP system combines prompts from different categories. The system can ask for verification followed by a request for additional information. Direct request for verification of every piece of extracted information is time consuming and can invoke irritation of the client. To avoid irritation, the information is sometimes verified

indirectly. Consider for example the combination of "You want to depart from Delft. At what time do you want to depart?" into "At what time do you want to depart from Delft?"

Misunderstanding
If the ASP system is unable to extract any information from the last utterance, it will ask for repetition. If misunderstanding occurs repeatedly, it will stop the dialogue and ask the caller if he wants to be connected to the human operator.

In the VIOS prototype about 100 different templates are used. Everyone of these templates has one or more open fields, which can be filled with names of railway stations or time expressions.

3 Utterances

When we want to simulate a caller effectively, we have to prepare answers to all possible requests of an ASP system. Therefore, we have to be able to identify all possible prompts of the system. Once the prompt is identified, a possible answer only depends on the global category of the system's request: verification, clarification, new info, or a combination of these. The precise linguistic representation is unimportant. Answers can be created for every category, by following a set of guidelines:

Verification
Appropriate answers are "yes" or "no".

New information
These requests can be answered by whatever new information is available. The client is allowed to present any information he thinks suitable, just like in human–human OVR dialogues.

Clarification
Clarifications do not occur frequently and can be handled by using another representation, if any is available.

Combination
The combination category can be problematic. When a wrong verification is combined or followed by a request for new information, there is no appropriate answer. The most appropriate answer to such a request is to restart the dialogue from the beginning or again provide the utterance from which the ASP system extracted the wrong information.

The Simcall system is based on a selection of train information dialogues from our OVR corpus containing over 5000 human–human public transport dialogues [Vark et al., 1996]. From these dialogues, we selected those utterances that can be used as answers to possible prompts. We illustrate the procedure with an example. We extracted the following utterances from the dialogue in figure 2:

- opening sentence: Hello, [name] is speaking, I like to travel from Hengelo to Schiphol,
- place of departure: from Hengelo,

goedemorgen reisinformatie	Good morning, travel information
hallo, met [name], ik zou graag van Hengelo naar Schiphol willen reizen	Hello, [name] speaking, I like to travel from Hengelo to Schiphol
ja en wanneer wilt U daar naartoe gaan?	yes, when do you want to go there?
nou, ik moet er een uurtje of half twee zijn	well I have to arrive about half past one
13.30 aankomen?	arrive at 13.30?
ja	yes
en is het voor vandaag?	and is it for today?
ja, natuurlijk	yes, of course

Figure 2: Example dialogue of the OVR corpus

- place of arrival: to Schiphol,
- combined: from Hengelo to Schiphol,
- time of arrival: about half past one,
- date: today,
- open slot: departure time.

4 Simcall

The Simcall system is composed of the following modules:

- a database containing all prerecorded utterances. These utterances are grouped according to the dialogue from which they originate. The database contains a total of 200 dialogues, selected from our corpus of 5000 human–human dialogues.
- a graphical user interface to facilitate interaction with the testbed (see figure 3). By clicking appropriate buttons a new session can be started. A dialogue can be selected along with the corresponding utterances. These buttons can be activated by labeled buttons.
- a connection module by which Simcall can connect to the ASP system. This module uses a special telephone card to interface the inquiry system directly. The connection module can record the complete dialogue if needed.

In every test trial, Simcall has a limited set of utterances to its disposal. This set of utterances belongs to the selected dialogue from the OVR corpus. Every utterance covering more than one information slot is also split into single slot utterances. Both the original utterances and the derived utterances are the only utterances the system is allowed to use.

The general strategy used by Simcall is to apply the original utterances whenever possible. In case of misunderstandings or clarifications, the derived single slot utterances can be used.

5 Experiments

Over the last 2 years, several experiments have been applied to the VIOS prototype or its precursors. These experiments lead the Simcall system as it exists nowadays.

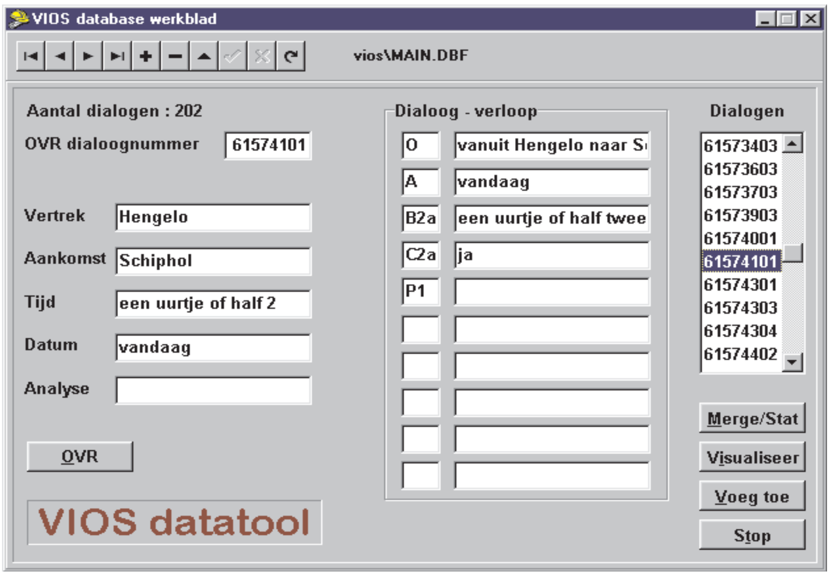

Figure 3: Graphical user interface of Simcall

5.1 Experiment 1

The database contains 200 dialogues with selected prompts from human–human dialogues. The VIOS system was called by members of the research team who were instructed to use only the recorded prompts belonging to a selected dialogue. Four students (two man, two woman) called the VIOS system 200 times.

The experiment proved that with use of the selected utterances it was possible to have a dialogue with the VIOS system. Only minor linguistic changes in the selected utterances were necessary.

5.2 Experiment 2

The students from the first experiment were replaced by a Simcall prototype. After selecting a dialogue all the relevant prompts are activated and presented as buttons in the user interface. The Simcall system was used to make 202 connections to VIOS. Figure 4 shows an example of information gathered during this experiment: a flowchart of how VIOS uses prompts to retrieve time information.

Miscommunication occurred in 23% of the dialogues. Without error correction 70% of the calls were handled successfully. Using error correction, 85% of the calls resulted in a correct travelling schedule.

5.3 Experiment 3

Using Simcall as it was used in experiment 2, a human operator is necessary to activate the relevant prompts by clicking the corresponding buttons. To disable the human operator an automated version under development. To function properly, the different prompts of the ASP system must be distinguishable automatically. Fortunately the VIOS system has only a limited number of prompts, which belong to one of two categories:

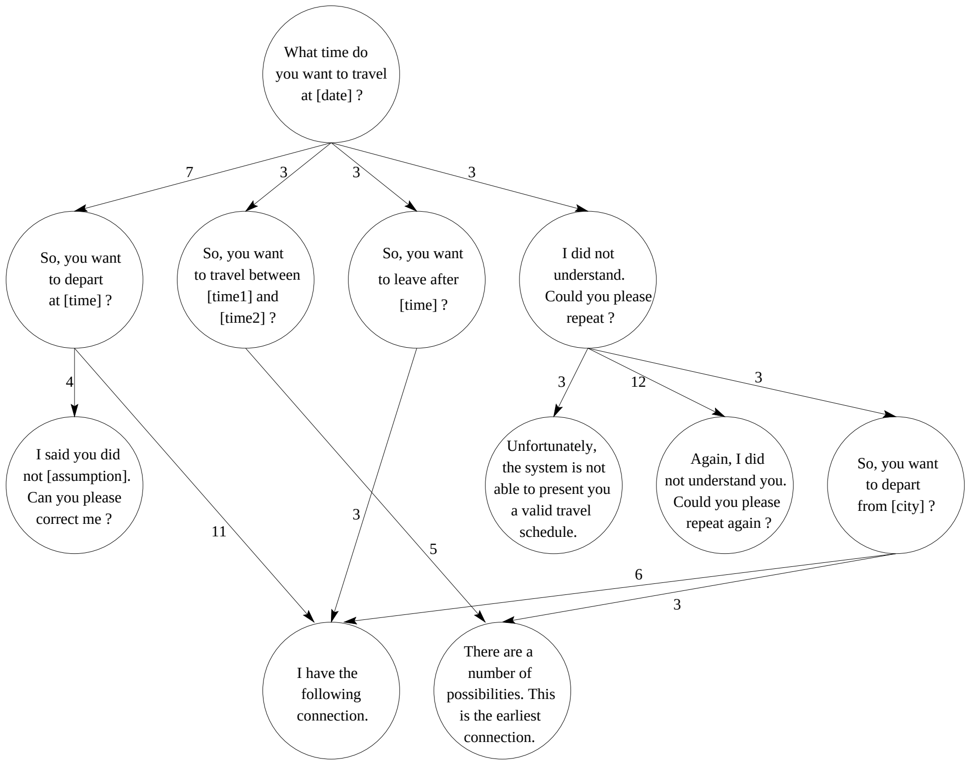

Figure 4: Structure how VIOS fills time slots

- **invariant prompts**
  These prompts are uniquely characterised by the length of the utterance. A tool is developed for automated real-time assessment of the length of the prompts.

- **concatenated prompts**
  These prompts are templates which are completed with information for place and time expressions. During the experiment, it was again shown that utterance length completely characterised the utterances, although detailed measurements were neeeded.

6 Conclusions

A semiautomated version of Simcall is presented. With use of that system it is possible to assess the performance of ASP systems. The great advantage of such a system is that the assessment of an ASP can be done in short time with a minimal of effort. Every ASP system is tested with use of the 200 recorded dialogues. Based on this dialogues an assessment scale can be developed. To prevent the set of 200 dialogues becoming part of the test set for ASP systems, the test dialogues have to be masked. For that reason the test set of 200 dialogues was enlarged by adding and manipulating the following operators: noise, pauses and nonverbal utterances, and interruptions. In the future, we will have to enlarge our database of linguistic expressions and improve masking.

Future work includes assessing the duration of all prompts accurately. To do this properly, we need a great number of test calls to record all prompts with as many different slots as possible.

Another item we will work on in the near future, is adding speech recognition to acquire a fully automated version of Simcall. As ASP systems generate a finite number of prompts using one speaker, it is possible to recognise prompts with a high accuracy as speech recognition does not have to be speaker independent and only has to recognise a small vocabulary. Additionally, VIOS does not manipulate the concatenated speech fragments, resulting in an identical acoustic representation for every occurrence of the same prompt.

References

[Aust et al., 1995] H. Aust, M. Oerder, F. Seide, and V. Steinbiss (1995), The Philips Automatic Train Timetable Information System, *Speech Communication*, 17:249–262.

[Oerder and Aust, 1994] M. Oerder and H. Aust (1994), A Realtime Prototype of an Automatic Inquiry System, in *International Conference on Spoken Language Processing*, pp. 703–706.

[Rothkrantz et al., 1997] L.J.M. Rothkrantz, W.A.Th. Manintveld, M.M.M. Rats, R.J. van Vark, J.P.M. de Vreught, and H. Koppelaar (1997), An Appreciation Study of an ASR Inquiry System, in *To appear in the Proceedings of the EuroSpeech 1997 Conference*.

[Vark et al., 1996] R.J. van Vark, J.P.M. de Vreught, and L.J.M. Rothkrantz (1996), Classification of Public Transport Information Dialogues using an Information Based Coding Scheme, in *Workshop Dialogue Processing in Spoken Language Systems*, European Conference on Artificial Intelligence, pp. 92–99, revised paper has also appeared in Lecture Notes in Artificial Intelligence 1236, pp. 55–69, 1997.

[Veldhuijzen van Zanten, 1996] G. Veldhuijzen van Zanten (1996), Pragmatic Interpretation and Dialogue Management in Spoken–Language Systems, in *Dialogue Management in Natural Language Systems*, 11th Twente Workshop on Language Technology, pp. 81–88, University of Twente.