Multilingual Speech Interfaces (MSI) and Dialogue Design Environments for Computer Telephony Services

Christel Müller and Thomas Ziem
Deutsche Telekom Berkom GmbH
Research Group Speech Systems and Computer Telephony
10589 Berlin, Goslarer Ufer 35, Germany
Tel. +49 30 3497 2310, FAX: +49 30 3497 2967, E-mail: c.mueller@berkom.de

Abstract

Today voice processing systems especially in the area of telecommunication are used by a wide range of customers in more and more countries. The most important facts for people who create applications based on voice processing systems are rapid prototyping, updating, modification and extension of dialogues as well as their standardised components.

At Deutsche Telekom research lab of speech systems and computer telephony a graphical user interface for designing computer telephony applications was developed as part of a CTI (Computer Telephony Integration) model on top of standardised interfaces for speech and other CT components in a client server architecture.

The approach to this CTI model ensures the independence of different resources e.g. ASR, TTS and other.

The most complicated part of this CTI model was the realisation of the resource management layer for multilingual speech interfaces because different ASR technologies do not support unified recognition functions and resource parameters.

The dialogue as a technological interface between user and voice processing system on one hand and dialogue as an assessment facility for voice technologies in research on the other hand require the improvement of dialogue design tools.

It is important to keep the balance between specific features of speech interfaces and restrictions of voice processing technology since both factors influence and enhance each other in the process of technological advance.

2. CTI Model

The CTI model shown in fig. 1 provides an optimal environment for the rapid user-oriented development and deployment of complex CT services with unified components integrated into communication networks.

Furthermore the integration of new speech technologies is supported with an independent resource layer.

The CTI model is user-centred and consists of the following main layers: user interface, application, resource management, resources and network.

1. Introduction

Existing graphical user interfaces do not include the integration of ASR functions in an appropriate manner where dialogue designers can easily handle vocabularies and grammars in different dialogue states.

Therefore the Telekom research group developed a dialogue design environment including unified interfaces for ASR and TTS.

Additionally a new feature which allows the application developer to design processes running in parallel was implemented at the graphical layer.

During the process of specification the main dialogue design criteria were integrated into the GUI, i.e. user experience, self interpretation, error robustness, flexibility and consistency.

User

Network

Resource

ResourceManagement

Application

User Interface

Fig. 1: User-centred CTI model
Results of Deutsche Telekom research were implemented into the application and the resource management layer of the CTI model. A short description of any layer is described below.

2.1. User Interface Layer

The wide range of user interfaces is divided into the CTI types of user groups:
- designer or integrator,
- provider and user of a CT service
with their different input and output modes.

2.1.1. Users of a service

- Closed user groups: Local groups, e.g. in a defined area, in a town, user with special needs
- Language dependent groups: e.g. French speaking users
- Network dependent groups: e.g. users in a mobile network
- Open or independent user groups: Multilingual users in different networks

2.1.2. Designer, Integrator, Provider

Today the process of designing and/or integrating a network based CT service is mostly carried out by developers or programmers of application generators rather than by the users themselves. Provider of services are telecommunication companies and large enterprises.

2.1.3. Interface Modes

<table>
<thead>
<tr>
<th>Input</th>
<th>Mode</th>
<th>Output Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>manual:</td>
<td>Keyboard:</td>
<td>PC, DTMF Telephone, Notebook, Fax,....</td>
</tr>
<tr>
<td></td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Touch Screen</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gesture</td>
<td></td>
</tr>
<tr>
<td>speech:</td>
<td>Microphone:</td>
<td>Telephone, Speech</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PC</td>
</tr>
<tr>
<td>visual:</td>
<td>Screen</td>
<td>Text, Picture</td>
</tr>
</tbody>
</table>

2.2. Application Layer

The application layer consists of the following parts:
- Graphical editor,
- Debugger,
- Simulator,
- Runtime,
- Console Management and Statistics.

The application layer and its functionality is mostly user adapted. While designers and integrators primary deploy graphical editor, debugger and simulator, users are only interested in reliable runtime versions and providers may prefer console management, analysis of statistical data and maintenance CT systems.

2.3. Resource Management Layer

The resource management layer describes the application software abstraction layer and the resource software abstraction layer which consist of a set of dynamic link libraries. The other important part is called resource software which controls the signal processing of the different resources. A standardised abstract interface is provided for each class of resource functionality.

2.1.4. Resources Layer

Resources with their drivers, firmware and hardware are moving towards standardised hardware busses and standardised software libraries. The following standardised resource classes are supported at the resource management layer: player, recorder, signal detector and generator, call channel, ASR, TTS and Fax. The administration resource is under development.

2.1.5. Network Layer

The network layer includes the different types of networks between resources and users, LAN/WAN, Intranet/Internet and mobile, satellite and wired network in telecommunication.

3. Dialogue Design Environment

The dialogue design environment developed by Deutsche Telekom is an approach to realise the described application layer of the CTI model. Special emphasis was paid to the graphical user interfaces which enables the designer to create user-oriented applications.
3.1. Graphical User Interface (GUI)

The editor of the GUI offers a range of dialogue functions divided into several dialogue actions. Each of these actions has its own parameter list which can be modified by adding, deleting or changing single items. The GUI is completely event driven.

Two dialogue functions „Automatic Speech Recognition“ and „Translate“ are configured for speech processing:

<table>
<thead>
<tr>
<th>ASR</th>
<th>TRANSLATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare Vocab</td>
<td>Speech</td>
</tr>
<tr>
<td>Recognise</td>
<td>S2T</td>
</tr>
<tr>
<td>Select Vocab</td>
<td>T2S</td>
</tr>
<tr>
<td>Train</td>
<td>Text</td>
</tr>
<tr>
<td>Verify Vocab</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2 Automatic Speech Recognition, Speech and Text Translation and Conversion

3.2. Dialogue Actions Running as Parallel Processes

Most of the CT services require processes running in parallel, like playing a prompt, searching for a special information in a database or enabling the user to interrupt these actions by pressing a key or speaking an utterance.

The visual projection of these processes was implemented at the graphical level in two steps:

- starting all dialogue actions as parallel processes which have only the exit event „Running“
- stopping these processes with a „Wait for Event“ action, which allows to define special exit conditions for all parallel processes.

4. ASR Resource Management

Speech recognition, speaker verification and speaker identification are today's capabilities of ASR technologies.

Language identification is under research.

The ASR resource management layer represents the link between the application layer, e.g. the dialogue action „Recognise“ and the function call in an ASR resource, e.g. „start_recognition“.

4.1. Unified ASR Interfaces

The ASR resource management has to achieve the goal of unified ASR interfaces and accordingly vendor independence. This includes the recognition functions as well as the vocabulary, grammar and parameter settings.

4.1.1. Basic Recognition Functions and Parameters

The goal of the existing ASR resource management layer is to realize the independence of basic recognition functions and a selected set of parameters [1]. The handling of these parameters not only includes the initial settings but also the modification of their returned values and a repetitive setting:

„Recognize(Group,nNumResults,RTC, OptArgs,TranInfo,Mode)“
„RetrieveRecognition(Group,ResulType, Results,TranInfo,Mode)“.

The „Talk over“ functionality was additionally included into the basic recognition functionality to stop the recognition process by a DTMF tone or a special spoken word or phrase.

First „talk over“ had the following parameters:

- Initial Time-out
- Final Time-out,
- Maximum Time Window,
- Energy Floor.

4.1.2. Specific or Extended Functionality and Variables

Specific or extended functionality cover the whole range of grammar descriptions and definitions of speech understanding, special training functions and the handling of large vocabularies.

The complexity of these different recognition functionalities requires a language and a technology independent approach.

5. Multilingual ASR Resources Management

To create multilingual unified speech interfaces at the resource management level there has to be developed a method of handling unified grammars.

Existing languages are mainly described by methods like BNF grammar.

To change or substitute grammars on the fly is not the scope of existing speech interfaces but has to be integrated in the future.

Another functionality is the different pronunciation in the grammar. From the research experiences this functionality can not be implemented in an unified ASR interface but has to be separated for each language.
Otherwise the description of adding, deleting, activating and deactivating variables as well as the production rules of the CFG can be formulated in a unified way [1]. The dynamic handling of vocabularies or word lists contains adding, deleting and modification.

5.1. Example

A multilingual ASR functionality is described in the following example:
The ASR dialogue action „Recognise“ is designed at the graphical level from which the transaction with an existing unified ASR resource starts.
The following parameters for this dialogue action are added:

- Continuous,
- InitialTimeout,
- FinalTimeout,
- MaxTimeWindow.

The application layer sends a request message to the resource management layer which is dispatched to the ASR resource layer for processing.
ASR resource answers this processing by generating a response message which is routed back to the application.
This ASR resource management described above can be used for any digits independent of the handled language.
First research work is done to provide language related components - grammars, lexicons, parser, ASR engine, TTS engine, and inference engine - in a modular interchangeable fashion [2].

6. Conclusions

The results at Deutsche Telekom research group has led to the conclusion that the unified ASR interface with extended functionality remains a difficult challenge. Especially the process of natural language understanding is far from implementation in dialogue design environments. However most researchers and first users agree that speech technology is ready for use now.
With well defined user interfaces and reasonable environments today’s real-time applications meet the needs of users.

References