A project of Speech Input and Output in an E-Commerce Application

Diamantino Freitas[1], António Moura[2], Daniela Braga[3], Helder Ferreira[1], João Paulo Teixeira[2], Maria João Barros[2], Paulo Gouveia[2], Vagner Latsch[1]

[1]Faculdade de Engenharia da Universidade do Porto,
{dfreitas, hfilipe, vagner}@fe.up.pt
[2]Instituto Politécnico de Bragança
{joaopt, moura, mjbarros, pgouveia}@ipb.pt
[3]Escola Superior de Educação – Instituto Politécnico do Porto
dbraga@ese.ipp.pt

Abstract. The present paper describes the work done during the last year in the development of European Portuguese (EP) speech recognition and synthesis channels for an Internet e-commerce application. The objective of this work was to develop the appropriate speech input and output, to enable the user to more comfortably use an e-commerce web site issuing commands and options under guidance of a menu system. The speech interface operation guidelines are briefly described in their principles and resulting specifications for the speech channels, namely, menu structure and operation dynamics, and in the following, the system software approach and the speech recognition and synthesis modules are presented. A discussion is done about some project trade-offs and results obtained so far. Future perspectives of the on-going work are also presented.

1 Introduction

Use of e-commerce applications on the Internet is steadily growing and motivates for advances in user interfacing that provide additional comfort and ease of use. It is a type of application in which, depending on the target area of e-commerce, vocabularies can be moderately short to allow the introduction of small speech recognition engines. In the present work the food supply area of e-commerce was the target and the work presented in this paper is co-ordinated with the general interface design, under the responsibility of a project partner.

The products categories in this field of business are structured by the IFLS standard[1] that provides a tree-type of organization with 3 levels. Some of the most typical problems arise from the presence of foreign words in product names, causing increased difficulty in speech synthesis, from the dynamics of the interface, in response to users commands, and from the acoustical noise and room conditions, the latter causing problems at voice pick-up for speech recognition. In the present approach a

[1] Institut Français pour le développement des Liens et Services Industrie-Commerce.

listen-and-select mechanism was adopted by the interface design partner employing speech prompts issued by the system whenever a user action is required or information is conveyed to the user. The adoption of a menu driven interface with natural numbering of items from lists, allowed the use of an isolated word recognition approach. The rather low-count in the global vocabulary for speech recognition together with a small perplexity at each menu, consisting mainly of the menu keywords together with integer numbers and operational commands, indicated a moderate difficulty for the recognition task in this aspect.

In the speech output channel, the present application posed a number of relevant problems, specially in the aspects connected to naturalness of output speech. Following the conclusions of the work of *COST 258*[2], the areas of quality at segmental and supra-segmental levels were addressed. At the segmental level, the diphone concatenation concept was adopted and two signal generation techniques were implemented. Firstly, a formant-based generator was produced in the direct sequence of existing know-how. Due to the limitations in signal quality typical of the formant-based approach, a second signal generator was started in the project and finalized by its end, this one based on time concatenation of original speech diphones, using a specially developed technique based on RELP-OLA[3] synthesis and a second diphone database, developed for the purpose, in the scope of a larger EP database development described elsewhere [1]. Not less important for synthetic speech naturalness are the supra-segmental level aspects. Text analysis and pre-processing for phonetic conversion and the generation of adequate prosodic contours for the speech signal, mainly in the f0 and segmental durations domains, were considered. For both processes, an *XML*-based[4] part-of-speech tagger (LAB205ML) was developed in order to supply contextual and structural information to each subsequent module in the speech synthesizer chain. The prosodic programming required a substantial number of studies including production of specific speech and linguistic databases. The main phrase categories used in the e-commerce system interface were identified and the corresponding prosodic patterns extracted, stylized and programmed.

Another general problem was the software interface with the operating system of the PC terminal under *Windows*. The *Microsoft SAPI 4.0*[5] system was adopted and a reasonably good compatibility with the browser software was achieved, together with a stable management of the input and output speech processes. An object-oriented programming approach was used along the work enabling the complex interpenetration of text information down to the signal generation level, for use when necessary.

2 The Voice Interface operation

[2] COST 258 – Naturalness of Synthetic Speech (http://www.unil.ch/imm/docs/LAIP/COST_258/cost258.htm)

[3] Residual Excited Linear Prediction – OverLap and Add.

[4] Extensible Mark-up Language.

[5] SAPI – Speech Application Program Interface.

The communication of the speech channels with the multimedia system of the PC and the browser application is achieved through the *SAPI 4.0* that implements a set of components and interfaces for communication with text-to-speech (*TTS*) and speech recognition (*SR*) systems [2][3].

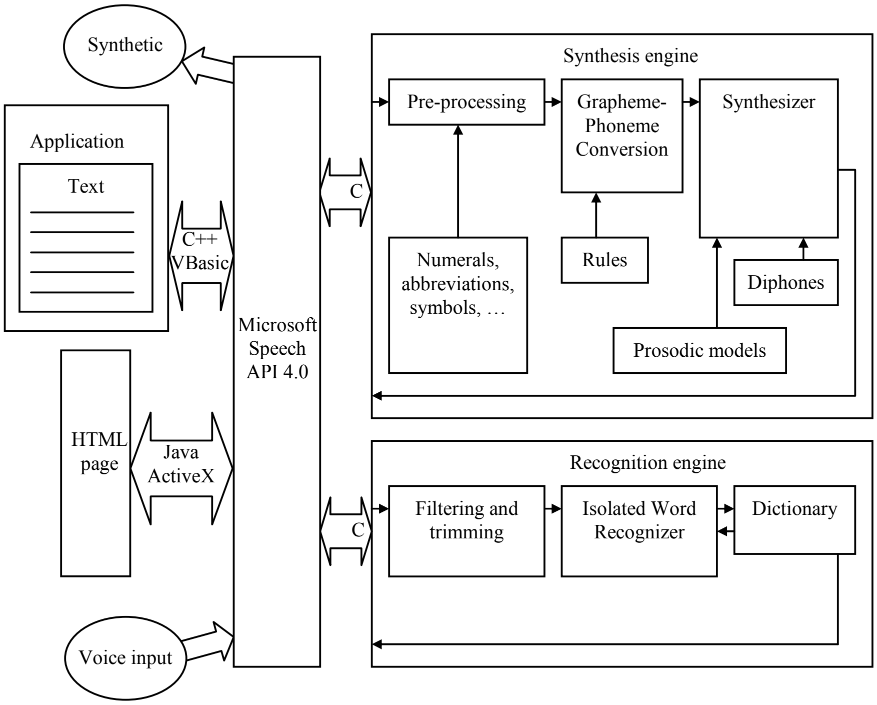

Fig. 1. General architecture of a SAPI compatible application.

Figure 1 illustrates the architecture of a generic application that employs *TTS* conversion and isolated word recognition (*IWR*).

The diagram can be interpreted at two levels: the speech channels level and the application level. The application with voice menu navigation uses both speech channels. These channels implement in a direct or indirect way the functions that the application requires at each phase. The central issue is to implement the channels software in a way compatible with most applications. The *SAPI* is placed between the application and system levels. The *TTS* e *SR* systems are called *engines* in the *API* and are installed in the operating system. Through the *SAPI* the (or each) application queries the available engines and chooses the preferred voice/language for *TTS* or the preferred *SR*. The implemented components belong to several complexity levels and it is up to the application to select the best at each moment. The lower the component level the more direct is the access to the engines.

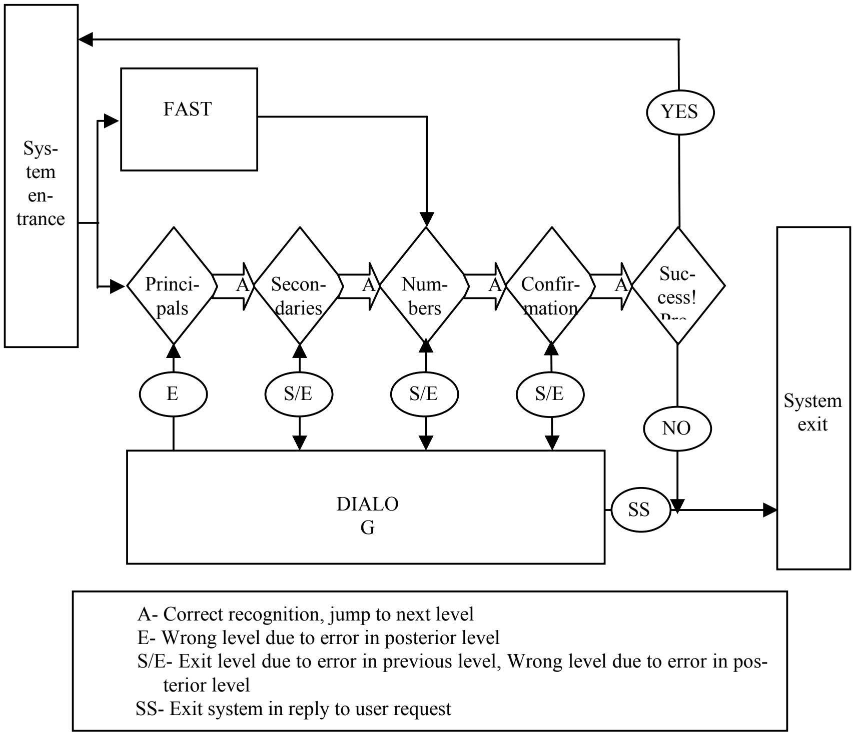
Sys-
tem
en-
trance
FAST
YES
Princi-
pals
A
Secon-
daries
A
Num-
bers
A
Confir-
mation
A
Suc-
cess!
E
S/E
S/E
S/E
NO
System
exit
DIALO
G
SS
A- Correct recognition, jump to next level
E- Wrong level due to error in posterior level
S/E- Exit level due to error in previous level, Wrong level due to error in posterior level
SS- Exit system in reply to user request

Fig. 2. The IFLS hierarchy and its interface

The 3 *API*s of lower level are: DirectTextToSpeech, DirectSpeechRecognition and Audio Object. All are based in COM[6] object programming [4]. In the first 2 APIs it is up to the engines to implement the interfaces and the objects. In the case of the Audio Object both the interfaces and objects are implemented already and ready to be used by the engines.

Common characteristics in both *TTS* e *SR* engines are their *dll*[7] shape that implements objects and interfaces. The first object needed is the *enumerator* that supplies the application information about a specific engine. After selection of an engine the *engine* object is created. Each object possesses its own characteristics of *TTS* or *SR* operation.

2.1 IFLS Structure and menu voice interface

[6] COM - Component Object Model.

[7] dll - Dynamic link library.

In order to clarify the requirements of the voice interface tasks, the IFLS-based system recognition structure is briefly presented. It is divided into levels, organised in a hierarchic way, with sequential access between levels, as can be seen in Figure 2.

3 The Speech Recognition Module

This section reports the development of the speech recognition module for isolated words of a restrict vocabulary including natural numbers. It's a speaker dependent system, which uses speaker adaptation techniques.

3.1 Architecture

The system is based on a discrete hidden Markov model approach. It is composed of the following parts: pre-processor (filtering and endpoint detection), features extraction, codebooks generation, array quantization, models training, adaptation and word recognizer. The recognizer internal architecture is presented in Figure 3 below.
Development of all modules was done in C programming language.

3.2 Recording of Speech Data Base

A one speaker database was recorded at 16 ksamples/sec, 16 bits, mono. The entire database was recorded with the same type of microphone in quiet conditions, with the microphone positioned at about 15 cms from the mouth, slightly below and to the side. Some trial recordings were done with different positions for testing and selection purposes. The recordings were done in 6 sessions, three at night and three during the day, distributed along a period of two months.

The database consists of a phonetically rich 15 minutes speech recording, suitable for the codebook generation, and an average of 100 sound samples (70 for training and 30 for testing) of each word to recognize in the system (natural numbers and vocabulary words). In order to achieve the best possible scores in word recognition, the database quality was an important point that was considered.

3.3 The Recognition System

Pre-processing: In order to remove low frequency noise (and DC baseline), all sound input signals need to be pre-processed. A 4th-order, IIR band-pass filter was introduced for the band 100Hz-7,8KHz. To eliminate silence segments before and after utterances, an endpoint detector algorithm was developed.

Features extraction: Energy, delta-energy, mel-cepstral, delta-mel-cepstral and delta-delta-mel-cepstral coefficients were extracted, making a total of 38 coefficients.

Codebook generation: The system codebook, with 128 elements, is calculated from the total of the 15 minutes phonetically rich recorded speech, after the features extraction. Array quantization: The sound signal was divided into 20 ms frames, 50% overlapping, and each frame was quantized on the calculated codebook. When the utterances are quantized the system can process the next stage.

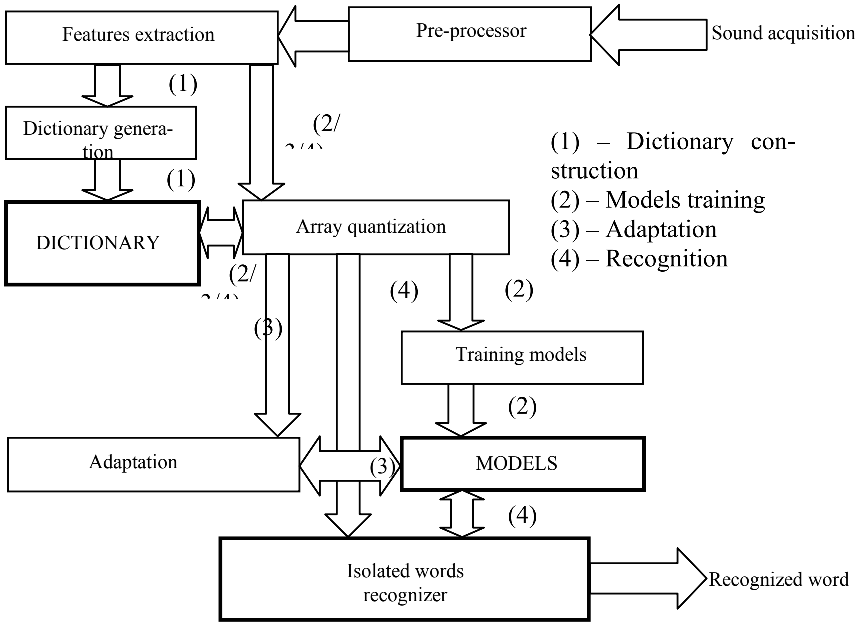

Fig.3. The Recognition System Architecture

Models Training: The training algorithm implemented is based on discrete hidden Markov models with 3 states per syllable. Adaptation: The adaptation algorithm implemented to adapt the recognizer to another user is also based on discrete hidden Markov models and the average duration of each word. Words Recognizer: The homogeneous and first order hidden Markov models were trained using the Baum-Welsch algorithm and also uses an estimate of the average duration of the word. The selection of the recognized word is done using a maximum likelihood estimator – MLE over the quantized signal and duration data.

Besides the *engine* object, the recognition engine requires a *gramma*r object. The engine contains the filtering and signal cut algorithms. The grammar object itself performs the recognition.

The recognition system works with a fair precision presenting some sensitivity to ambient noise, what is to be expected in the actual technology. Use of a close-talk microphone, refinement of the algorithms and models with a larger database will lead to a higher robustness of the system.

4 The Speech Synthesis System

4.1 Text Pre-Processing

The text-to-speech system presented in Figure 4 below receives the text to be synthesized through the *SAPI* and creates an internal object whose structure comprehends a full range of text characteristics and parsing information, from initial text down to the phone level. This structure is then filled-up by the subsequent processes. The text is afterwards passed through a pre-processing phase where mainly numerals, abbreviations and other symbols are converted into a readable full-length text form. The first step for conversion is to detect the category of the expression (number, abbreviation, etc). This module is still under development. Meanwhile a basic set of categories is automatically detected (integer numerals for instance), the more sophisticated categories still require mark-up. Manual mark-up is possible in the scope of the present application although not advisable as a method. The next step is the conversion itself that is already fully developed, and comprehends most of the possible categories of expressions. In this stage the text is already split into smaller units, paragraphs or phrases, and buffered to speed up the process.

4.2 Linguistic Analysis

A morph-syntactic analysis of the pre-processed text is done, in order to prepare an adequate prosodic manipulation [5]. For this purpose a rule-based grammar was created, and is presently in the programming phase, in PROLOG, on top of a morphologic analyzer from PRIBERAM. It provides a text parsing function into phrasal groups, with some morphologic disambiguation (a preliminary version). As they become increasingly available in the on-going development work of this project, the results of the morph-syntactic analysis, are used for the selection of the prosodic pattern [6] of intonation from a set of models that were determined in specific studies for the present application. Proposed prosodic categories in the model library include declarative, interrogative and imperative modes with a comprehensive range of relevant sub-modes. This development is described in more depth elsewhere [7].
The results of the, above referred, conversion and analysis, allow mark-up of the text using a mark-up language, called Lab205ML, created specially for this purpose. This language comprehends prosodic mark-up of phrasal groups. The use of mark-up is interesting for transmission of parsing and text structure information between successive modules of the TTS system. It also allows forced selection of text conversion modalities or prosodic patterns when needed.

This can be interesting in the present type of applications where the texts to be read by the system are limited in type, in structure and in communication modality.

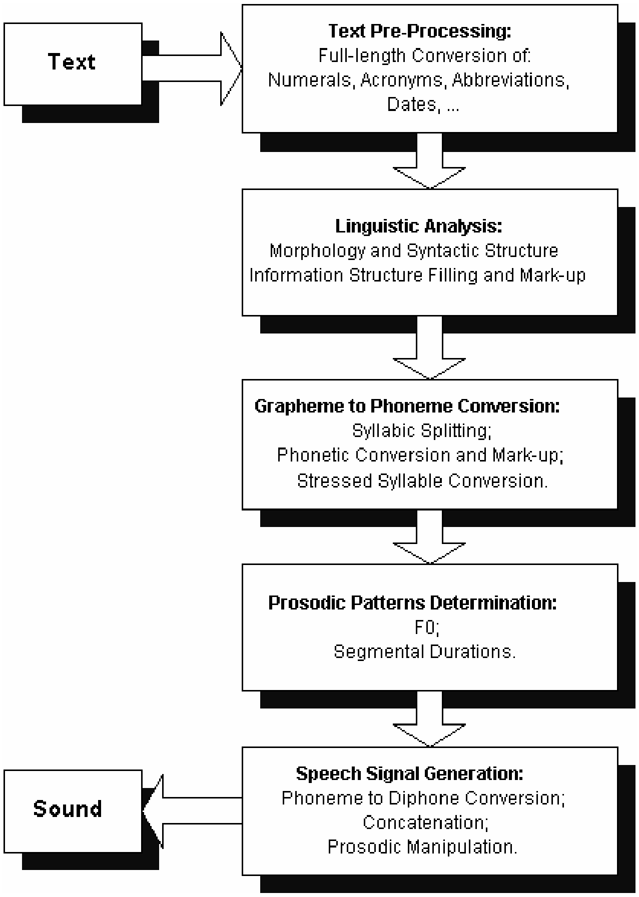
Text
Text Pre-Processing:
Full-length Conversion of:
Numerals, Acronyms, Abbreviations,
Dates, ...
Linguistic Analysis:
Morphology and Syntactic Structure
Information Structure Filling and Mark-up
Grapheme to Phoneme Conversion:
Syllabic Splitting;
Phonetic Conversion and Mark-up;
Stressed Syllable Conversion.
Prosodic Patterns Determination:
F0;
Segmental Durations.
Speech Signal Generation:
Phoneme to Diphone Conversion;
Concatenation;
Prosodic Manipulation.
Sound

Fig.4. The Text-to-Speech System Architecture

For this purpose a mark-up development system was created to assist manual or semi-automatic text mark-up to be used by the contents editor.

4.3 Text conversion, prosodic pattern selection and signal generation

Each phrase will subsequently pass through the operation of grapheme-phoneme conversion (phonetic transcription). This conversion maps the set of characters into one set of symbols that represent the sound units (phonemes).

The implemented algorithm converts a sequence of graphical ASCII[8] characters, into SAMPA[9] phonetic characters. For example, the phrase: “Uma frase declarativa.”

[8] ASCII- American National Standard Code for Information Interchange.

is converted into: /_um6fraz@d@kl6r6tiv6_/. The conversion algorithm, that also performs syllabic division, stressed syllable detection, and some co-articulation transformations, inserts marks of beginning of phrase, word and syllable and marks of stressed syllable, into the converted text sequence. The mark of stressed syllable, for instance, will be used later in the prosodic processing phase for time placement of word accent.

The sequence of phonemes with prosodic information, indicates which diphones to concatenate for database search and collection and the results passed to the signal-processing phase by means of the selected time units concatenation synthesizer. This unit is selectable between 2 types: a 4-formant with improved L-F model excitation source [8] and improved mouth radiation filter, and one RELP-OLA concatenation device with original speech diphones. The synthesizer in any case is prepared for processing the speech signal according to the prosodic f0 and duration patterns prepared from the linguistic analysis. For the intonation (f0) pattern, the Fujisaki model [9] was used for manipulation. For segmental durations a statistical model was built using artificial neural networks. The speech signal then obtained is buffered and delivered to the operating system sound sub-system.

5 Present Status of Results, Conclusions and Future Perspectives

The set of speech channels has been integrated with the Internet application of e-commerce to produce a prototype whose functionality has been proved. The prototype is presently under tests and final adjustments to enter the phase of user tests. In the speech recognition channel the speaker dependent approach produces some recognition errors with other speakers and the adaptation scheme must be used. A much larger database is presently under preparation to circumvent this limitation together with evolution of the adaptation scheme. A next version of the recognizer will be based on a semi-continuous HMM approach for improved resolution.

The speech synthesis channel is presently capable of reading the text materials of the application and most of the general texts as well. Some improvements in the text pre-processing are being done at the levels of automatic classification of expressions for full-length conversion (special numerals, etc) and phonetic conversion (treatment of rules exceptions).

The morph-syntactic analysis device is being continuously improved with programming and tuning of rules (and exceptions). For the moment the automatic operation of the prosodic module is only capable of identifying some types of phrasal groups and defaults to the basic ones. The prosodic behavior of the synthesizer is therefore still not fully natural. Extrapolating from the near past the impact of improvements on naturalness it can be concluded that naturalness will still increase very much with the conclusion of the mentioned developments.

Regarding segmental quality the two signal generators behave differently. The formant based version is capable of a good phonetic realization due to the diphone approach and the improved source, but lacks naturalness at the spectral level due to

[9] SAMPA- Speech Assessment Methods Phonetic Alphabet.

the limitations of the employed model. The RELP-OLA synthesizer produces a better segmental quality, but prosodic manipulation is more difficult. Users opinions favour the RELP-OLA device for segmental quality, but formal tests need to be performed in near to real situations for solid conclusions to be taken. At the signal level, objective tests are under way using a spectral distance approach.

A formal evaluation of the errors at the different text processing phases is currently under way. The method used is based on large manually tagged texts, different from the ones used for development, that are used as references for calculation of error rates.

6 Acknowledgements

The authors wish to acknowledge and thank: AdI (Agência de Inovação - MCT) for the support granted to the project where this work is inscribed. COST 258 – Naturalness of Synthetic Speech, for the great scientific impact in the present work. PRIBERAM, for licensing their morphological analyzer for EP (http://www.priberam.pt).

7 References

1. "Phonetic Events from the Labeling of the European Portuguese Database for Speech Synthesis, FEUP/IPB-DB", J. P. Teixeira, D. Freitas, D. Braga, M. J. Barros, V. Latsch, Proceedings of "EUROSPEECH 2001", Aalborg, Denmark, September 2001.
2. Microsoft Speech API 4.0. http://www.microsoft.com/iit/onlinedocs.
3. Developing a text-to-Speech Engine. http://www.microsoft.com/iit/onlinedocs.
4. Dr. GUI on Components, COM and ATL. http://msdn.Microsoft.com/library.
5. "Estudio de Técnicas de Processado Linguístico y Acústico para Sistemas de Conversión Texto-Voz en Español basados en Conactenación de Unidades", Lopez, Eduardo, Doctoral Thesis, Universidad Politécnica de Madrid, (1993).
6. Cruz-Ferreira, Madalena; "Intonation in European Portuguese", in Hirst, D.; Di-Cristo, A.; Intonational Systems, Cambridge University Press, (1998).
7. "Correlation between Phonetic factors and linguistic events regarding a prosodic pattern of European Portuguese: a practical proposal", D. Freitas, D. Braga, M. J. Barros, V.Latsch, J. P. Teixeira, Proceedings of "ICSP2001 – International Conference on Speech Processing", Seoul, August 2001.
8. Childers, Donald G. "Speech processing an synthesis toolboxes," John Wiley &Sons, inc. (1999).
9. Fujisaki, Y. et al.; Computing Prosody, Spring New York, USA, ISBN 0-387-94804-X, (1997), ch. 3, pp. 27-40.