First published in:
IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 14, NO. 2, MARCH 2006

Flexible Speech Translation Systems

Tanja Schultz, Alan W. Black, Stephan Vogel, and Monika Woszczyna

***Abstract*—Speech translation research has made significant progress over the years with many high-visibility efforts showing that translation of spontaneously spoken speech from and to diverse languages is possible and applicable in a variety of domains. As language and domains continue to expand, practical concerns such as portability and reconfigurability of speech come into play: system maintenance becomes a key issue and data is never sufficient to cover the changing domains over varying languages. In this paper, we discuss strategies to overcome the limits of today's speech translation systems. In the first part, we describe our layered system architecture that allows for easy component integration, resource sharing across components, comparison of alternative approaches, and the migration toward hybrid desktop/PDA or stand-alone PDA systems. In the second part, we show how flexibility and reconfigurability is implemented by more radically relying on learning approaches and use our English–Thai two-way speech translation system as a concrete example.**

***Index Terms*—Multilinguality, portability, speech translation, system deployment.**

I. Introduction

RESEARCH on speech translation has shown that it is possible to build systems that translate spontaneously spoken utterances from one language to another. To handle the challenge of ambiguity introduced by spontaneous speech, most systems introduce semantic constraints by limiting the domain of discourse, thereby reducing the number of suitable interpretations. For many applications, constraining the domain (hotel reservation, scheduling, travel planning, etc.) is quite acceptable and can provide practical translation devices. Over the years, we have developed a large number of such systems for many languages, domains, and platforms. These efforts have shown that acceptable performance can be obtained for spontaneous speech input, but also that practical concerns such as portability and reconfigurability become increasingly important. The two main challenges are in maintaining the existing system and obtaining enough data for each new language and domain. How can the cost of data collection and programming effort be contained? What translation approaches should be used? How do we move from desktop prototypes to portable devices with limited computational power and memory footprint? How should such portable devices be integrated and how should they communicate to deliver translation capability effectively?

Manuscript received June 20, 2004; revised June 16, 2005. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Grant "Mobile Speech-to-Speech Translation for Military Field Applications" (formerly called "Babylon") and LASER-ACTD under Grant "CMU Thai Speech Translator for Multilingual Coalition Conversation." The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Giuseppe (G. E.) Riccardi.

The authors are with the Interactive Systems Laboratory, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: tanja@cs.cmu.edu).

Digital Object Identifier 10.1109/TSA.2005.860768

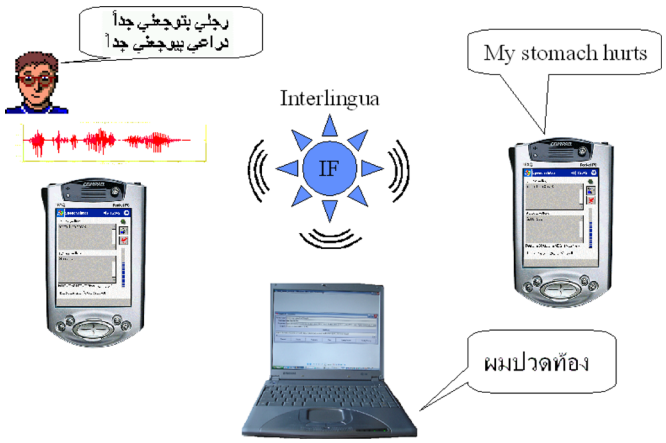


Fig. 1. Multilingual speech-to-speech translation. Typical speech translation scenario: an English-speaking participant speaks into a handheld device. Speech gets 1) recognized, 2) parsed into an intermediate textual meaning representation (Interlingua), 3) sent wirelessly to the recipients 4) generated into Arabic/Thai, and 5) synthesized on the recipient's device.

II. A Scalable and Flexible System Design

This section introduces each component, i.e., automatic speech recognition (ASR), speech synthesis (TTS), natural language understanding (NLU), as well as machine translation (MT) and its integration into a scalable, flexible, and reconfigurable speech translation system. Fig. 1 shows a typical scenario for our speech translation system, which supports a typical multilingual speech translation task. A native English medical doctor wants to communicate with an Arabic or Thai patient. The doctor speaks English into his personal handheld device. The speech is recognized and parsed on this device (English ASR and parsing). An intermediate textual meaning representation of the sentence, the interlingua, is sent wirelessly to the patient's device (the same handheld, a second handheld, or a laptop/desktop). The recipient's device generates audible output in the target language Arabic or Thai (generation and TTS), and processes the recipient's spoken response (Arabic/Thai ASR and parsing). In order to support this scenario, each component of the speech translation system needs to be scalable enough to run efficiently on platforms ranging from handheld devices to desktops.

A. System Deployment

When it comes to the actual use and, thus, the practical deployment of a speech translation system, smaller platforms are more convenient to the user. Unfortunately, the larger or at least the more computationally capable the platform is, the better the translation can be. We have taken these conflicting requirements into account and designed a system that can take advantage of both. We built a system that runs on small platforms, such as a PDA, as well as on larger platforms such as desktops or laptops.

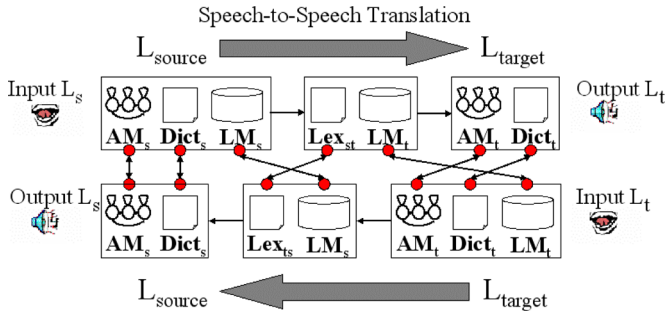


Fig. 2. Resource sharing in a S-2-S system.

To combine the benefits of both solutions, the system's architecture supports a seamless integration of both platforms based on wireless communication.

1) Multiple Platform Communication: The communication between the devices is by wireless (Bluetooth or WiFi 802.11). In order to keep the bandwidth limited, we pass a textual representation between the devices rather than recorded speech. Transferring audio to a single central server for recognition, translation, and synthesis would require higher bandwidth and a centralized infrastructure. Our interlingua-based machine translation strategy uses an interchange format (IF) as representation that is passed in a textual representation. In our scenario, the sender's language is recognized and analyzed on the sender's device while the receiver's language is generated and synthesized on the receiver's device. This scenario also naturally supports one-to-many translations, in which the sender's speech input is simultaneously translated into different output languages on several recipients' devices. The overall architecture is, however, not restricted to pass IF representations between the devices. Other configurations using speech recognizer output, n-best lists, or lattices are also supported. We chose IF since it fits well with a one-language-per-device scenario.

2) Small, Mobile Platforms: It should be explicitly stated that small footprint devices such as PDAs are, at least for the foreseeable future, not simply lower powered desktop machines. They are limited mostly by battery power, which does not allow them to double their processing power every 18 months with Moore's law. To save energy, handheld devices are limited in their CPU speed and memory. Most PDA-class machines do not have floating-point processors (though typically offer slow emulation of such). In order to run ASR, NLU, MT, and TTS engines efficiently such devices it is necessary to reengineer them. Our system uses different engines when deployed on PDAs, but the models are trained from the same data as their large counterparts (see next section).

3) Resource Sharing Between Components: In order to conserve memory, to minimize the developmental effort, and to reduce the maintenance costs, knowledge sources can be shared across components. Fig. 2 indicates all knowledge sources of a two-way speech translation system that can be shared across components. For example, the design and implementation of the phone inventory is a major concern for both speech recognition and speech synthesis. Therefore, we decided to use the same phone inventory for the handheld and desktop ASR engines as well as the TTS. All three can share the same dictionary resources.

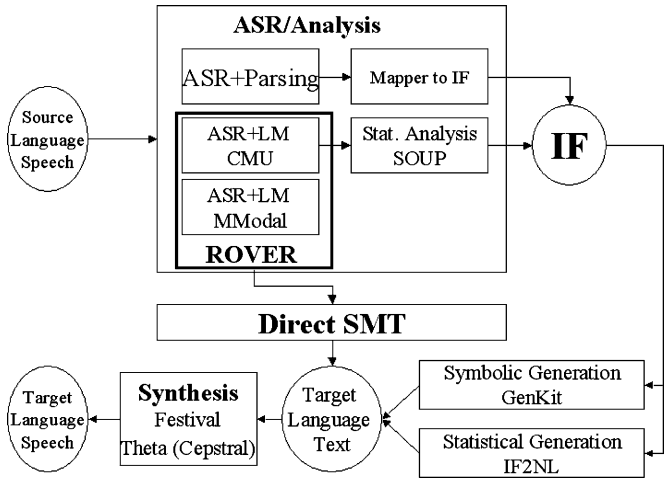


Fig. 3. System architecture.

We can bootstrap the M*Modal handheld recognizer (see Section II-B1) using the phonetic alignments of CMU's desktop recognizer. Both recognizers can also exchange statistical language models (SLMs) in the National Institute of Standards and Technology (NIST) ARPA format. Furthermore, the grammars developed for interlingua-based translation can be read in for context free grammar-based speech recognition. Since language models are memory consuming, sharing them between ASR and statistical MT (SMT) becomes a valuable aspect.

B. System Architecture and Components

To support a scalable system that handles complex translation tasks, we designed a flexible system architecture as shown in Fig. 3. The architecture allows the development in distributed teams, supporting decoupled implementation and improvement of ASR, TTS, NLU and MT. The current system runs on Windows, WinCE, and Linux operating systems and supports two recognizers, two translation strategies, and two synthesis engines. The spoken input in the source language is passed on to the ASR/analysis component. This component supports two recognition strategies. One uses statistical n-gram language models $(\text{ASR} + \text{LM})$, the other uses grammars to recognize and parse the input into an interlingua representation $(\text{ASR} + \text{Parsing})$. The statistical parser SOUP [40] allows for allows for parsing the ASR + LM output. The recognition can be performed with either CMUs or M*Modal's speech engine. The ASR output is then passed on to one of two different machine translation components: IF-based or direct SMT. In case of the interlingua-based translation strategy, the target language textual representation is generated from IF. The SMT translates the textual representation of the source language directly into the target language. Finally, the target language text gets synthesized using either the Festival Speech Synthesis system or Cepstral's small footprint synthesis engine Theta.

1) Automatic Speech Recognition: The two recognizers used in our system are CMU's Janus JRTk recognizer [1], [2], and M*Modal's recognizer [3]. On a desktop or laptop, both recognizers may either run interchangeably or combined. Both

TABLE I
GRAMMAR VERSUS N-GRAM LANGUAGE MODEL

	N-gram	Grammar	ML-Grammar
English – WER	21.7%	24.9%	25.9%
English – SER	47.4%	44.1%	43.2%
English – RT	0.7	0.48	0.67
German – WER	26.5%	24.0%	25.0%
German – SER	54.4%	42.2%	42.7%
German – RT	1.29	0.48	0.61

produce a flat hypothesis string and a recognition word lattice graph, annotated with confidence scores for combination by ROVER or confusion networks [4]. When running on a handheld device, the M∗Modal recognizer uses specialized compact models. Depending on the amount of available resources at runtime, both recognizers can be based on grammars as well as on SLMs. The M∗Modal recognizer allows expanding the domain on the fly in several ways. In case of an SLM system, a system designer or a front-end user can add words to language-model classes (pronunciations will be generated through a built-in grapheme-to-phoneme module) or adapt the language model by providing additional sentences. In case of a grammar-based system, the rules can be edited and modified on the fly. The easiest cases are to add new flat top-level expressions and to expand lists of open-class terminals, such as medication names. The runtime requirements for the M ∗ Modal recognizer are between 3 MB for small grammar-based systems and 50 MB for large SLM-based systems. The speech recognition performance on the handheld for English in a medical domain using a trigram LM roughly corresponds to the one on a desktop at 1/3 real time, with a 10% relative increase in word error rate.

In early bootstrap phases where no or too little data are available for building SLMs, grammar-based recognition is extremely helpful. As more data become available, they are used to increase grammar coverage and learn transition weights in the grammars. As the amount of language model data increases (for instance by transcribing data logged while the system is used or by mining internet resources) using statistical language models becomes more practical than the labor-intensive work of increasing grammar coverage. This is especially important since systems with very large grammars require more resources than those based on statistical language models. Since the grammar experts often include grammar rules for paraphrases of training sentences, the resulting grammars may have better coverage than the initial corpus. Therefore, formerly created grammars are typically unrolled to generate additional data for SLM training.

Table I compares the performance of grammars to SLMs for the JRTk recognizer for German and English in a tourist assistant and navigation domain in terms of sentence error rate (SER), word error rate (WER), and real time (RT) behavior (computed on a 1.13-GHz Pentium III Mobile Processor). The test data consists of 820 utterances spoken by 11 German and nine English native speakers, resulting in 31-min speech. The vocabulary has about 2500 words for German and 2000 for English. The language model was trained on 260 000 words for English and 9100 words for German. One hundred ninety-eight rules for the context-free grammar had been manually designed to cover the English domain, 132 rules were created for German. The bilingual acoustic model was trained on 15 h English Broadcast News, 40 h English Verbmobil, and 60 h German Verbmobil.

Comparison of the English monolingual results (the first two columns) shows that the statistical language model outperforms the grammar in WER, but performs worse in case of the SER. For German, the grammar outperforms the statistical model for WER and SER, mainly due to the small training corpus. In terms of real-time factors, the grammars are roughly twice as fast as the n-gram models. The German SLM system is slower because of the poor coverage. These results indicate that grammars are preferable over statistical language models if the domain and training corpus is restricted and real-time performance is of key concern. The last column shows that folding monolingual grammars into one multilingual (ML) grammar in combination with multilingual acoustic models yields reasonable performance with small real time losses. The resulting multilingual speech interface implicitly performs language identification (error rate on English is 3.6%, on German 1.8%) [5].

2) Machine Translation: Our system architecture supports two translation engines, an interlingua-based and a statistical MT component.

Interlingua-based machine translation: The interlingua-based MT analyzes a sentence into a language independent semantic representation (IF) using an analysis grammar and generates the target sentence using a generation grammar. To build a domain-specific translation system requires the design of an interlingua and the development of analysis and generation grammars. The IF developed by CMU has been expanded to encompass concepts in both the travel and medical domains, as well as many general-use or cross-domain concepts. It has proven to be portable to various domains and languages [6]. The design of an interlingua has to balance expressive power and simplicity [7]. The specification at the argument level attempts to distinguish between domain-dependent and domain-independent sets of arguments, in order to better support portability to new domains. The interlingua also has to be simple enough so that grammar developers can work independently on different languages at different sites. A simple and a more complex example of utterances with corresponding IFs are shown here.

1) *Thank you very much.*
   `c:thank`
2) *On the twelfth we have a single and a double available.*
   $\texttt{a : give} - \texttt{information} + \texttt{availability} + \texttt{room}$
   $(\text{time} = (\text{md12}), \text{room\_type} = (\text{single} \,\&\, \text{double}))$

The advantage of semantic grammars is that the parse tree that results from analyzing an utterance is very close to its final semantic interpretation. A disadvantage is that new grammars have to be developed for each domain, although some low-level modules such as those covering time expressions can be reused across domains. Strong advantages of interlingua-based MT are that: 1) it abstracts away from variations in syntax across languages, providing potentially deep analysis of meaning without relying on information pertinent only to one particular language pair and 2) the users can be given a paraphrase in their own language, which can help verify the accuracy of the analysis and be used to alert the listener to inaccurate translations. With respect to a flexible system architecture, interlingua-based MT

modules for analysis and generation can be developed by monolingual persons, with additional knowledge only of the second "language" of the interlingua. This is important in translation between small languages where the chances to find a bilingual expert are low.

Statistical direct translation: Statistical machine translation has been advocated by the IBM research group in the early 1990s [8] and since then has become a very active research field. The approach is based on Bayes' decision rule: given a source sentence f_1^J of length J, the translation e_1^I is given by

$$\hat{e} = \arg\max_{e_1^I}\{\Pr(e_1^I|f_1^J)\} = \arg\max_{e_1^I}\{\Pr(f_1^J|e_1^I)\Pr(e_1^I)\}.$$

Here, $\mathrm{p}(e_1^I)$ is the language model of the target language, typically a 3-gram, and $\mathrm{p}(f_1^J|e_1^I)$ is the translation model. A number of different translation models, also called alignment models, are used [8]–[11]. The argmax operation denotes the search problem. The SMT decoder applied to these experiments uses a beam search based on dynamic programming [12], [13].

As it is the case with any statistical system, more data usually results in better performance. One of the arguments in favor of knowledge-based systems like the interlingua-based MT system is that statistical MT can often not be used in limited domain translation tasks, as the bilingual corpora are usually extremely limited. However, an advantage of the SMT system is that it is easy to incorporate other knowledge sources like general-purpose bilingual dictionaries, large monolingual corpora to estimate the language model parameters, and out-of-domain bilingual corpora. This typically involves data filtering, formatting, and cleaning, but this is less work intensive than translating a significant amount of domain-specific data or writing translation grammars.

Comparing IL-MT and SMT: To see if the statistical approach to translation is applicable to domain-limited speech translation, where the bilingual corpus is of very limited size, we evaluated the SMT system on the NESPOLE! translation task [14] and compared it to an existing IL-MT system [6] which was developed as part of the NESPOLE! project. To make the results from the SMT system comparable to those from the interlingua-based MT system, the same data was used for training the alignment model as was used to develop the analysis and generation grammar. Two-thousand four-hundred twenty-seven German–English sentence pairs with 11 236 German and 11 729 English tokens were used to train the statistical system. The vocabularies contained 1196 German and 1010 English words. For the SMT system, HMM alignments [10] from German to English and from English to German were generated and used to extract a word-to-word statistical lexicon and phrase-to-phrase translations from the bilingual corpus.

A human evaluation was carried out for the comparative evaluation. Several dialogs, adding up to 194 sentences were used as a test set. Translation was from German into English. Six evaluators were presented with the German turn and the two translations. Grading was done on a three-point scale: "Good" (translation gives the required information and is easy to understand); "Okay" (translation gives useful information, even if it is syntactically not correct); "Bad" (translation is missing, gives no useful information, or is misleading).

TABLE II
EVALUATION RESULTS FOR IL-MT AND SMT

	Good	Okay	Bad	Acceptable
IF	18.9	36.3	44.8	55.2
SMT	40.3	22.7	37.0	63.0

The evaluation results are given in Table II. The numbers for "Good," "Okay," and "Bad" translations are given in percentages, with counts accumulated over the six evaluators. Comparing the results from both systems shows that statistical translation is at least competitive, yielding comparable translation quality in significantly less development time.

3) Speech Synthesis: Current technology in speech synthesis is concentrated on *concatenative speech synthesis*, selecting appropriate subword units from a natural speech database and concatenating them to form new utterances. Previous techniques, such as formant synthesis, required significant knowledge and skill to construct for new languages. The building of voices in new languages is based on the work described in the FestVox project [16], which offers tools and techniques for building new synthetic voices. The techniques we designed provide working voices for the Festival Speech Synthesis System, a free software system that runs on most platforms [17]. Much of the quality of a concatenative unit selection synthesizer depends not just on the engine and selection algorithms but also on the coverage of the database the units are selected from. Festival requires significant resources to run, and unit selection databases can easily require hundreds of megabytes. In order to offer the ability to run voices on much smaller platforms, Cepstral, LLC developed a new engine, called Theta [18], which uses voice compression and pruning techniques to allow the same high-quality synthesis on a handheld. The quality of the results directly depends on the effort involved. Although a reasonable domain targeted voice can be built fairly quickly for Festival, small footprint general voices take more effort in design, labeling, and tuning.

III. SYSTEM FLEXIBILITY AND RECONFIGURABILITY

In this section, we discuss how to reduce the developmental effort for practical deployment of speech translation systems. While part of the discussion is abstracted away from a particular target language, we also demonstrate how the current system supports speech translation in a realistic setting and how it supports the rapid adaptation to a new target language. We report quantitative scores in terms of development time and performance numbers for single components. These numbers are based on our two-way English–Thai speech translation system, which was developed in the framework of Babylon, supported by DARPA and Laser-ACTD. The end-to-end system was also evaluated within the Babylon framework by MITRE [19].

A. Language Peculiarities

We have used our tools and techniques for a large range of human languages, and this has made them more robust and general. We have frequently encountered new aspects that we had not yet catered for. We have gained significant experience dealing with many language groups, from the major European

and Asian languages, to various resource-limited indigenous languages. This has allowed us to address various language phenomena that can make speech recognition, translation, and synthesis hard, such as tonality (Chinese, Thai), various inflection systems including rich morphology (German, Korean, Serbo–Croatian, Turkish), lacking word segmentation (Japanese, Chinese), different phonological structures (simple mora-based as in Japanese versus complex consonantal clusters as in German), and various writing systems (Roman, Cyrillic, Korean, Devanagari, Chinese).

Within a speech translation system, there are sometimes solutions that might not be possible in other speech and language systems. For example, in conventional written Arabic [20], vowels are not included in the written script. It is a nontrivial task to automatically predict which vowels are missing. There has been work on Arabic speech recognition that ignores vowels, but it is difficult to use such implicit vowel systems for speech synthesis. For our two-way English–Arabic speech translation system, we use a closely phonetic internal representation with full vowel information. We can transform this representation into Arabic script, but not the reverse. A similar simplification is made in our Thai system [21]. Normal Thai script does not contain spaces between words. We have a word segmentation algorithm [22], and within the system we only use the segmented representation. This makes the translation and synthesis easier, without requiring each process to resegment the Thai character string.

Spoken language can differ quite significantly from written language. Such differences can vary from language to language. For example, Modern Standard Arabic (MSA) is a well-defined written language and can also be used for more formal speech such as news broadcasts. It allows a common communication over a wide range of dialects in the Arabic-speaking world. However, MSA is not a standard spoken language and in designing speech translations systems it is important to cater for *spoken* language. In Thai, there is also a distinction between the written form and spoken language. For example, sentence final particles are influenced by the gender of the speaker. Thus, it is important to collect in-domain *conversational* speech to model the differences in natural spoken language.

B. Rapid Model Building for ASR

We have accumulated considerable experience in language adaptation techniques and our recognizer has been successfully applied to more than ten languages [23], which cover a variety of characteristics as described above. Based on our multilingual data collection efforts GlobalPhone (GP) [24], we train a global, language-independent phone set and then adapt the acoustic models to new languages. Our rapid adaptation techniques enable us to bootstrap acoustic models in a new language on limited training data. We first collect general domain speech data for the new language to create general acoustic models. For the latest developments in Arabic and Thai language, we collected read speech from about 100 speakers. Since read speech does not require costly transcription work, this kind of data collection can be done very quickly and efficiently. For Thai, a single person collected 20 h of GlobalPhone style data in Thailand and prepared it for acoustic model training in three weeks.

TABLE III
WER ON GP, PHONE SET, AND PRONUNCIATION VARIANTS

System / #Acoustic Models	500	1000	2000
Baseline	16.0	15.2	14.6
Enhanced phone set	16.0	-	14.4
Pronunciation variants	15.6	14.8	**14.0**

Phone set: The Thai phoneme set consists of 21 consonantal phonemes, 17 consonantal cluster phonemes, and 24 vowels. Each vowel can carry one of five tones: low, mid, high, rising, and falling. When investigating the impact of tone information, we found no performance gain [25]. Therefore, we focused on phone sets without tone features. Our baseline phone set consists of 42 phones: 21 consonantal, and 21 vowel phones. We split the 17 consonantal cluster phonemes into two separate phones. For comparison, we built an enhanced phone set by adding the 17 cluster phonemes, resulting in 59 phones. The results in Table III show no significant gains by using the enhanced phone set. In a third experiment, we took the smaller phone set and built a dictionary with multiple pronunciation variants to handle the most common pronunciation variation effects (on average 1.1 variants per word). This gives up to 4% relative improvement. We ran the model training on 80 native speakers of the Thai GlobalPhone data. For testing we used 1181 utterances spoken by eight different speakers. The language model was built on news articles and gives a trigram perplexity of 140 and an out-of-vocabulary (OOV) rate of 1.4% based on an 8k vocabulary.

Pronunciation dictionary generation: The dictionaries used in the experiments described above rely on manually transcribed pronunciations of a base vocabulary. Using this model, we built a statistical letter-to-sound model to predict the pronunciation of new words, and hand corrected errors [26]. By iterating this method, we built a dictionary with 8000 entries. Pronunciation variants were generated by applying a set of (manually created) rules to this dictionary. However, in rapid adaptation of ASR to new languages, we cannot assume that pronunciations of a base vocabulary exist, nor that native experts are available for hand corrections. Recently, grapheme-based models for ASR have been proposed [27], which back up results indicating that pronunciation variants should not be explicitly modeled through phone string variations but implicitly by the use of single-pronunciation dictionaries [28] and parameter sharing across phonetic models [29].

A purely grapheme-based dictionary cannot capture the fact that depending on the context 1) the same grapheme might be pronounced in different ways or 2) that different graphemes might be pronounced the same way. The traditional clustering procedure is able to deal with the effects of 1). In order to handle case 2) and make the best use of the available training data, we allow sharing across different center graphemes by applying our flexible tree tying scheme in which a single decision tree is constructed for all substates of all graphemes [30]. We investigated grapheme-based ASR on various languages and found that grapheme based systems perform significantly worse for languages with poor grapheme-to-phoneme relation such as English, but achieve comparable or even better results for languages with a closer grapheme-to-phoneme relations such as Spanish, Russian, and German [31], [32].

TABLE IV
WER ON GP GRAPHEME VERSUS PHONEME

System (500 quinphone models)	WER
Phoneme-based (16k Gaussians)	16.0
Grapheme-based (16k Gaussians)	26.4
Grapheme-based + Flexible Tying (9k Gaussians)	18.3

TABLE V
INITIALIZATION ACROSS LANGUAGES

System (500 quinphone models)	HR	ML7
Grapheme-based	26.4	27.0
Phoneme-based (incl. Pron. Variants)	15.6	16.5

Table IV compares the results of the phoneme-based with the grapheme-based approach for Thai. The baseline is the previously described system based on 500 quinphone models using the smaller phone set without pronunciation variants. The grapheme-based system uses 500 "quingrapheme" models based on 63 Thai graphemes. The pronunciation dictionary is constructed by splitting the written word into its single character components. We also applied one simple rule, as in the Thai writing system, the grapheme for a vowel is sometimes written in front of the consonant even when it is spoken after the consonant. The results in Table IV show that the straightforward approach without sharing any parameters across graphemes does not work well for Thai. This is a consequence of several peculiarities of the Thai writing system: 1) one phoneme can be represented by many graphemes, even by two nonconsecutive graphemes, e.g., in the word "และ" (/lae/) the vowel (/ae/) is represented by two graphemes "แ ะ" which enclose the consonant "ล" (/l/), 2) one grapheme can represent different phonemes, and 3) a special written tag "์" (karan) suppresses the pronunciation of the tagged grapheme. The sharing of parameters across different grapheme models using flexible tree tying resolves these problems.

Initialization across languages: We bootstrapped our acoustic models from initial alignments generated by a very small acoustic model trained on 4 h transcribed Thai spontaneous speech dialogs of Hotel Reservations (HR) from Thailand's National Electronics and Computer Technology Center (NECTEC). The experiments in Table V show how much performance would be sacrificed if no Thai data was available for bootstrapping. We compared the bootstrap from the Thai HR models with a 7-lingual acoustic model (ML7) derived from the GlobalPhone project (trained on Chinese, Croatian, French, German, Japanese, Spanish, and Turkish). Table V displays a very reasonable loss when using the ML7 over the Thai HR models.

Rapid adaptation to Babylon: Finally, we adapted the best GP-based acoustic model to the Babylon domain of spontaneous medical dialogs between American doctors and Thai patients. For this purpose, we recorded a very limited set of speech data from ten native Thai speakers. Prompts were designed which include word forms typically occurring in spontaneously spoken Thai speech in medical dialogs. For adapting the acoustic models, we used 2433 utterances from eight speakers. The test set consists of 322 utterances from two speakers. The trigram language model has a perplexity of 41.8 with an out-of-vocabulary rate of 0.48%. The experiments were all performed on a fully continuous 3-state HMM recognizer with 500 quinphone models and 32 Gaussians per state.

TABLE VI
WER ON BABYLON CORPUS

System/Adaptation	GP only	GP+Bab MLLR	GP+Bab Mixed	Bab only
Baseline	24.6%	21.6%	20.6	21.5%
Enhanced phone set	23.1%	22.5%	-	22.6%
Pronunciation variants	23.7%	22.6%	**18.6**	19.6%

TABLE VII
TRAINING CORPUS STATISTICS

	43k Corpus		61k Corpus	
	Thai	English	Thai	English
Sentences	43,040			61,487
Words	345,773	325,212	456,479	440,879
Vocabulary	7540	7818	9662	9111

To adapt the acoustic models using this very limited training material, we investigated four schemes:

1) train acoustic models based on GlobalPhone training data only (GP only);
2) use the Babylon training material to MLLR-adapt the GP models (GP + Bab MLLR);
3) combine the training material of both corpora, weighting the Babylon material by a factor of 2 (GP + Bab Mixed),
4) use the GP models for initial alignments, but then completely retrain based on Babylon only material (Bab only).

The third and the fourth scheme include a reclustering of the decision tree. Table VI shows the performance for the different phone sets and dictionary variants. The results indicate that using a larger amount of general training material in combination with a limited amount of specialized training material to retrain including a reclustering of the decision tree is the best strategy.

C. Rapid Model Building for MT

While IF-based MT requires experts to manually design and implement the concepts, statistical MT permits automatic building of a translator. However, a bilingual corpus is needed to train the translation model. If no such corpus exists for the domain, some data collection is required.

We report on our speech translation experiments on Thai–English translation in the medical domain on two training sets: a corpus of 43k sentences pairs (22k provided by DLI and 21k collected by Mobile Technologies Inc.) and a corpus of 61k sentence pairs (35k from DLI, 26k from Mobile Technologies Inc.). Table VII shows the statistics of both corpora. The test set consists of 130 English sentences, for which the transliteration and the recognizer output for four different speakers were available. The vocabulary of the test sentences is well covered by the training data. The first three columns in Table VIII give the word error rates for the four speakers based on our ASR and the 3-gram perplexities for the language models built on the 43k and the 61k corpus, respectively.

The translation models were trained on the bilingual corpora, the language models were built using the Thai part of

TABLE VIII
EVALUATION RESULTS ON TEST SET

	WER	PP		NIST		BLEU	
		43k	61k	43k	61k	43k	61k
Text	0.0	34.0	31.6	4.29	5.99	17.6	29.4
ASR-spk1	4.5	35.4	33.1	4.11	4.74	15.8	22.8
ASR-spk2	9.0	36.9	35.3	3.86	4.41	15.4	20.4
ASR-spk3	9.2	38.7	36.6	3.76	4.37	13.0	20.0
ASR-spk4	14.8	39.0	37.7	3.81	4.35	13.6	18.4

TABLE IX
TRAINING DATA SELECTION EVALUATION

	NIST	BLEU
5k sentence selection	3.76	7.5
10k sentence selection	4.84	16.9
20k sentence selection	5.79	26.3
30k sentence selection	5.86	27.4
40k sentence selection	5.92	28.0

the corpora. For Thai, word segmentation was done as a preprocessing step. The original transcription (text) and the ASR output of each speaker were translated with both setups. The results, in terms of NIST and BiLingual Evaluation Understudy (BLEU) scores are given in Table VIII. We see a significant improvement by adding more text data for training. Translation quality degrades gracefully when going from text input to speech input, and the translation performance scales nicely with the recognition performance when comparing for the different speakers.

Training data selection: It has been shown that more data will give better results. However, as translating a collection of domain-specific sentences into a foreign language is time and cost consuming, it would be valuable to know which sentences in the corpus are of greater importance and, thus, should be incorporated first. Another reason to select more informative sentences for manual translation is the limited resources at runtime since training the models on more data also requires more memory at runtime.

More important data simply means the data, which gives best translation quality. A first target would, therefore, be to cover the vocabulary. The first selection criterion is formulated as: from all remaining sentences in the corpus select the one for which the ratio between number of new words and number of words in the sentence is largest. However, as translation quality depends strongly on the use of entire phrases, we extend the selection criterion to also include the coverage of n-grams. Experiments have shown that including bigrams in the selection process gives a significant improvement over using unigrams, while going beyond bigrams does not seem to help.

To test our selection criterion we trained systems on the 5k, 10k, 20k, 30k, and 40k most informative sentences from the 61k corpus. Table IX shows the results from translating the test set (text) based on two systems, one tuned toward the NIST score, one tuned toward the BLEU score. The results show that we have better NIST scores with a set of 10k well chosen sentences than with 43k randomly selected sentences. Even with only 5k sentences, we approach the 43k performance. We see also that with smaller corpora we can bring up the NIST score faster than the BLEU score. We get high vocabulary coverage fast, but good phrase coverage requires more data.

D. Rapid Model Building for TTS

We have developed general techniques for building synthetic voices in new languages [16] that can be run on the free software Festival Speech Synthesis System [17] be converted to run on Cepstral LLC's commercial Theta engine [18] that runs on handheld devices.

In building high-quality unit selection synthesis, it is necessary to collect sufficient natural speech to cover the intended phonetic and prosodic variation in the target language. When developing voices for languages without significant linguistic resources, we must take a generic approach that does rely on such knowledge. As described in more detail in [33], we select sentences that are optimized for phonetic coverage, from general text corpora. As we want the voice talents to read the sentences properly, the selected prompts should be as easy as possible to say. The less discrepancies between the prompts and the actual recordings the more we can depend on automatic labeling.

The basic selection process involves first finding sentences which are easy to say and have less chance of pronunciation error. We can further restrict this to disallow sentences containing homographs. We have discovered other worthwhile constraints, particularly to remove foreign words/names, especially if the selection corpus is newspaper text. Foreign words are likely to have unusual pronunciation, especially if our speaker is multilingual. When no lexicon of foreign names is available for a language, we use word frequency to limit our selection process. Thus, first we build word frequency lists and then select sentences that only contain high-frequency words.

For Thai, because we did not at first have a word segmentation system, nor a lexicon, we initially selected sentences based on letter tri-gram frequency. This gave us a basic phonetic coverage corpus to start recording. Later, following [34], we added domain specific sentences and selected these for word and phonetic coverage for the medical domain. A third set of prompts was handcrafted covering for example general greeting, numbers, and dates. It is worth noting that as this is an automatic process the resulting prompts may still not be ideal. They will tend to select for most varied phonetic coverage and often contain unusual word sequences, including typos. Thus, we still prune the resulting selections based on human judgment.

Building comprehensive lexicons for new languages can be a substantial task. As described in [26], first about 300 representative words are chosen and hand transcribed with their appropriate phoneme sequence. A statistical letter-to-sound rule model [35] is then trained from this data, which allows predicting pronunciations for new words. Next, we submit the most frequent words to the model, correct them manually where necessary, and add them to the lexicon. After adding a few hundred words, the letter-to-sound rule model is retrained. By repeating these steps, we converge on a set of words and letter-to-sound rules that pronounce arbitrary words in the language well. We have tested this technique on English and German with good results, and also applied it to Nepali, where no computational pronunciation lexicon had existed before.

Once the prompt set is recorded by the voice talent, it is automatically labeled using speaker-specific acoustic models, and forced alignment. Then, a cluster-based unit selection voice is

built [36]. Hand correction of labels can be worthwhile to improve quality.

Evaluation of voices, especially in new languages is hard. Although objective measures can be calculated for speech synthesizers for text analysis, lexicon and letter-to-sound rule coverage and accuracy, ultimately native listeners must be involved. Three levels of sentence type may be generated and listened to by 5–10 native speakers:

1) phonetic confusable words, following the Diagnostic Rhyme Test [37];
2) semantically unpredictable sentences (SUS) [38], where templates are used to generate sentences from simple part of speech classes;
3) in domain sentences, which are checked for intelligibility.

These tests are designed to be diagnostic, in other words, to identify lexical and labeling problems which can then be fixed. The results of applying these evaluation criteria are described for our Arabic work in [39].

E. System Evaluation

As with any complex system, it is not just the components that have to be evaluated, but also the whole system. For real users of a speech translation system, it is necessary that it presents an intuitive interface, and that they understand the system sufficiently to make an efficient communication aid. In many cases, one side of the conversations will involve a trained user who has used the system before (e.g., the doctor) while the other is new to the system. The conversations are likely to be led by the more expert user.

As part of the DARPA Babylon/CAST program, our English/Thai Doctor/Patient system was one of the participants in an evaluation with real American doctors and monolingual Thai speakers trained to act as patients. Although the number of participants was insufficient to produce significant quantitative results, a number of valuable observations were made. First, feedback is important to users so that they can judge if the system's translation conveyed the right message. To reduce confusion, we included the synthesized playback of the recognition output. Thus, the speaker could immediately identify when errors were occurring. This audible output was invaluable as a communication aid, since in some cases the screen was not visible to participants, and/or they had limited reading ability. Second, the time from speaking into the system to achieving the translated synthesized output is critical for multilingual conversations mediated by machines. Even when approaching real time, it is intrusive in the conversation flow. Thus, external hand signals for quick communication feedback helped significantly to make the conversations more fluent. In summary, carrying out evaluations in realistic scenarios we discovered that aspects of the interface, such as microphone accessibility and reliability of users using push-to-talk become significant.

IV. Conclusion

This paper describes our strategies to overcome the major limits of today's speech translation systems, namely the problem of system maintenance and the challenge of ever changing domains and/or languages resulting in a lack of appropriate data. The implemented flexible system architecture allows for easy component integration, resource sharing across components, comparison of alternative approaches, and the migration toward hybrid desktop/PDA or stand-alone PDA systems. The experiments and results carried out on our English–Thai two-way speech translation system show how flexibility and reconfigurability can be achieved by more radically relying on learning approaches. The recent system evaluation with real users indicates that the core technology in speech translation has now reached a level where user interface issues make a difference.

Acknowledgment

The authors wish to thank P. Charoenpornsawat, M. Eck, G. Flaherty, R. Frederking, C. Fügen, S. Hewavitharana, K. Lenzo, L. Mayfield-Tomokiyo, K. Peterson, S. Stüker, S. Suebvisai, V. Tespresit, and J. Zhang for their contributions.

References

[1] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, and M. Westphal, "The Karlsruhe Verbmobil speech recognition engine," in *Proc. ICASSP*, Munich, Germany, 1997, pp. 83–86.

[2] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *Proc. ASRU*, Madonna di Campiglio, Italy, 2001, pp. 214–217.

[3] M. Finke, J. Fritsch, D. Koll, and A. Waibel, "Modeling and efficient decoding of large vocabulary conversational speech," in *Proc. Eurospeech*, Budapest, Hungary, 1999, pp. 467–470.

[4] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Comput. Speech Lang.*, vol. 14, no. 4, p. 373, 2000.

[5] C. Fügen, S. Stüker, H. Soltau, F. Metze, and T. Schultz, "Efficient handling of multilingual language models," in *Proc. ASRU*, St. Thomas, VI, 2003, pp. 441–446.

[6] A. Lavie, L. Levin, T. Schultz, C. Langley, B. Han, A. Tribble, D. Gates, D. Wallace, and K. Peterson, "Domain portability in speech-to-speech translation," in *Proc. HLT*, San Diego, CA, 2001, pp. 82–86.

[7] L. Levin, D. Gates, A. Lavie, and A. Waibel, "An interlingua based on domain actions for machine translation of task-oriented dialogues," in *Proc. ICSLP*, Sydney, Australia, 1998, pp. 1155–1158.

[8] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, 1993.

[9] D. Wu, "Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora," in *Proc. IJCAI*, Montreal, QC, Canada, 1995, pp. 1328–1335.

[10] S. Vogel, H. Ney, and C. Tillmann, "HMM-based word alignment in statistical translation," in *Proc. COLING*, Copenhagen, Denmark, 1996, pp. 836–841.

[11] F. J. Och and H. Ney, "Improved statistical alignment models," in *Proc. ACL*, Hong Kong, China, 2000, pp. 440–470.

[12] S. Vogel, Y. Zhang, F. Huang, A. Tribble, A. Venogupal, B. Zhao, and A. Waibel, "The CMU statistical translation system," in *Proc. MT-Summit IX*, New Orleans, LA, 2003, pp. 402–409.

[13] S. Vogel, "SMT decoder dissected: word reordering," in *Proc. Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003, pp. 561–566.

[14] A. Lavie, C. Langley, A. Waibel, F. Pianesi, G. Lazzari, P. Coletti, L. Taddei, and F. Balducci, "Architecture and design considerations in Nespole!: A speech translation system for E-commerce applications," in *Proc. HLT*, San Diego, CA, 2001, pp. 31–34.

[15] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Association for Computational Linguistics*, Philadelphia, PA, 2002, pp. 311–318.

[16] A. Black and K. Lenzo. (2000) Building Voices in the Festival Speech Synthesis System. [Online] Available: http://festvox.org/bsv.

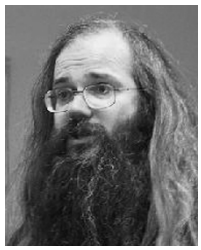
[17] A. Black, P. Taylor, and R. Caley. (1999) The Festival Speech Synthesis System. [Online] Available: http://festvox.org/festival.

[18] Cepstral, LLC, "Theta: Small Footprint Text-to-Speech Synthesizer," Cepstral, LLC, Pittsburgh, PA, 2004.
[19] J. Aberdeen, S. Condon, C. Doran, L. Harper, B. Oshika, and J. Phillips, *DARPA Cast Final Rep.*. Bedford, MA: The MITRE Corp., 2004.
[20] A. Waibel, A. Badran, A. Black, R. Frederking, D. Gates, A. Lavie, L. Levin, K. Lenzo, L. Mayfield Tomokiyo, J. Reichert, T. Schultz, D. Wallace, M. Woszczyna, and J. Zhang, "Speechalator: two-way speech-to-speech translation on a consumer PDA," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 369–372.
[21] T. Schultz, D. Alexander, A. Black, K. Peterson, S. Suebvisai, and A. Waibel, "A Thai speech translation system for medical dialogs," in *Proc. HLT*, Boston, MA, 2004, pp. 263–264.
[22] P. Charoenpornsawat, B. Kijsirikul, and S. Meknavin, "Feature-based Thai unknown word boundary identification using winnow," in *Proc. APCCAS*, Chieng Mai, Thailand, 1998, pp. 547–550.
[23] T. Schultz and A. Waibel, "Language independent and language adaptive acoustic modeling for speech recognition," *Speech Commun.*, vol. 35, no. 1–2, pp. 31–51, 2001.
[24] T. Schultz, "GlobalPhone: a multilingual speech and text database developed at Karlsruhe University," in *Proc. ICSLP*, Denver, CO, 2002, pp. 345–348.
[25] S. Suebvisai, P. Charoenpornsawat, A. Black, M. Woszczyna, and T. Schultz, "Thai automatic speech recognition," in *Proc. ICASSP*, Philadelphia, PA, 2005, pp. 857–860.
[26] S. Maskey, A. Black, and L. Mayfield Tomokiyo, "Optimally constructing phonetic lexicons in new languages," in *Proc. ICSLP*, Jeju Island, South Korea, 2004, pp. 1227–1230.
[27] S. Kanthak and H. Ney, "Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition," in *Proc. ICASSP*, Orlando, FL, 2002, pp. 845-1–845-8.
[28] T. Hain, "Implicit pronunciation modeling in ASR," in *ISCA Pronunciation Modeling Workshop*, 2002.
[29] M. Saraclar, H. J. Nock, and S. Khudanpur, "Pronunciation modeling by sharing Gaussian densities across phonetic models," *Comput. Speech Lang.*, vol. 14, pp. 137–160, 2000.
[30] H. Yu and T. Schultz, "Enhanced tree clustering with single pronunciation dictionary for conversational speech recognition," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 1896–1899.
[31] M. Killer, S. Stüker, and T. Schultz, "Grapheme based speech recognition," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 3141–3144.
[32] B. Mimer, S. Stüker, and T. Schultz, "Flexible tree clustering for grapheme-based speech recognition," in *Proc. Elektronische Sprachverarbeitung*, Cottbus, Germany, 2004.
[33] J. Kominek and A. Black, "The CMU ARCTIC speech databases for speech synthesis research," Lang. Technol. Inst., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-LTI-03-177. [Online]. Available: http://festvox.org/cmu_arctic., 2003.
[34] A. Black and K. Lenzo, "Limited domain synthesis," in *Proc. ICSLP*, Beijing, China, 2000, pp. 411–414.
[35] A. Black, K. Lenzo, and V. Pagel, "Issues in building general letter to sound rules," in *Proc. 3rd ESCA Workshop Speech Synthesis*, Jenolan Caves, Australia, 1998, pp. 77–80.
[36] A. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Proc. Eurospeech*, Rhodes, Greece, 1997, pp. 601–604.
[37] J. Logan, B. Greene, and D. Pisoni, "Segmental intelligibility of synthetic speech produced by rule," *J. Acoust. Soc. Amer.*, vol. 86, no. 2, pp. 566–581, 1989.
[38] C. Benoit, M. Grice, and V. Hazan, "The SUS test: a method for the assessment of text-to-speech synthesis intelligibility using semantically unpredictable sentences," *Speech Commun.*, vol. 18, pp. 381–392, 1996.
[39] L. Tomokiyo, A. Black, and K. Lenzo, "Arabic in my hand: small-footprint synthesis of Egyptian Arabic," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 2049–2052.
[40] M. Gavalda, "Soup: A parser for real-world spontaneous speech," in *New Developments in Parsing Technology*, H. Bunt and J. Carroll, Eds. Norwell, MA: Kluwer, 2004.

Tanja Schultz received the German Masters in mathematics, sports, and education science from the University of Heidelberg, Heidelberg, Germany, in 1990 and the M.S. and Ph.D. degrees in computer science from University Karlsruhe, Karlsruhe, Germany, in 1995 and 2000, respectively.

She is an Assistant Research Professor in the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, and heads the speech recognition group at the International Center on Advanced Communication Technologies (InterACT). Her research activities center around multilingual speech processing, with a particular area of expertise in rapid portability of speech recognition and speech translation systems to new languages.

Dr. Schultz was awarded with the FZI price for an outstanding Ph.D. in 2001 for her work on language independent and language adaptive acoustic modeling and the ISCA Best Paper Award in 2002.



Alan W. Black received the B.Sc. degree (Hons) in computer science from Coventry University, Coventry, U.K., in 1984, the the M.Sc. degree in knowledge-based systems from Edinburgh University, Edinburgh, U.K., in 1986, and the Ph.D. degree in computational linguistics from Edinburgh University in 1993.

He is an Associate Research Professor in the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA. He previously worked in the Centre for Speech Technology Research, Edinburgh University, and before that at ATR in Japan. He is one of the principal authors of the Festival Speech Synthesis System, the FestVox voice building tools, and CMU Flite, a small footprint speech synthesis engine. He is also Chief Scientist and cofounder of the for-profit company Cepstral, LLC.

Stephan Vogel received the Diploma in physics from Philips University Marburg, Marburg, Germany, and the M.Phil. degree from the University of Cambridge, Cambridge, U.K.

He is a Researcher at the Language Technologies Institute, Carnegie Mellon University (CMU), Pittsburgh, PA, where he heads the statistical machine translation team. Before coming to CMU, he worked for several years at the Technical University of Aachen, Aachen, Germany, on statistical machine translation, and also in the Interactive Systems Lab at the University of Karlsruhe, Karlsruhe, Germany. He had been working on several text and speech translation projects, including GALE, TIDES, Nespole, EuTrans, and Verbmobil.

Monika Woszczyna received the M.S. degree in physics (Dipl.Phys) and the Ph.D. degree in computer science (Dr.Ing.) from Karlsruhe University, Karlsruhe, Germany.

She is Head of Language Development at Multimodal Technologies, Inc., Pittsburgh, PA. She is also an Adjunct Faculty Member of the Language Technologies Institute, Carnegie Mellon University (CMU), Pittsburgh, PA. While at CMU, she worked in the Interactive Systems Laboratories coordinating the integration for the international speech-to-speech translation project, C-STAR-II.