Proceedings of 20th International Congress on Acoustics, ICA 2010

23-27 August 2010, Sydney, Australia

XML based notation for rule-based text-to-Speech systems

Dr.-Ing. Shahla Sehhati, Sehhati@cs.tu-berlin.de

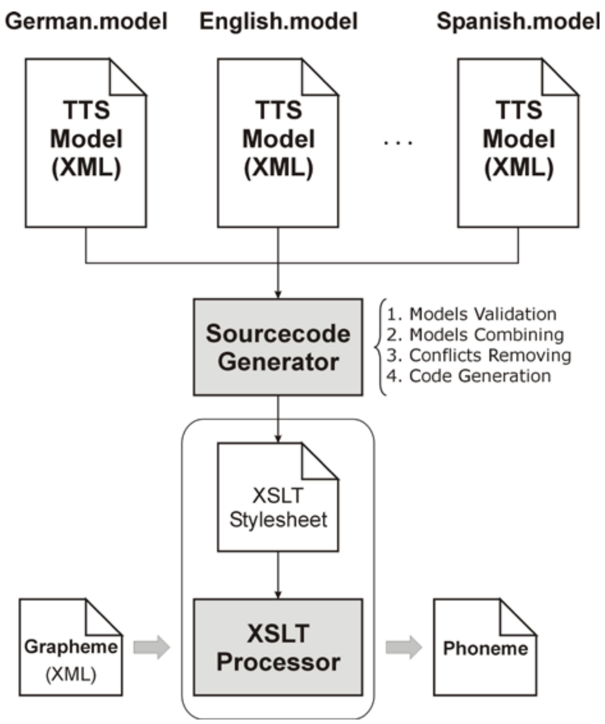
Technical University of Berlin, Germany

ABSTRACT

The usage of the Extensible Markup Language (XML) is not to be excluded in today's Text-To-Speech (TTS) systems any longer. XML does not come into application only for the internal data structures and to the modularity of the systems, but also is consulted for the communication of the client server architectures. However in past works the advantages of XML are not used for the step of the transcription. This aspect represents frequently an in itself locked module, which does not or only heavily permit an exchange of the transcription rules. But just rule-based multilingual TTS systems would profit from the integration of different sets of transcription rules. A standardized notation would simplify also the exchange between research groups. The available contribution suggests an XML based notation for rule-based TTS systems, with which a quantity of context sensitive rules can be described for the grapheme phoneme transcription system independently. As Proof of Concept typical applications are illustrated by the XML notation.

APPROACH OF THE PROCEDURES

The user is interested in to be able to enter and to evaluate comfortably a set of rules for the automatic transcription. In this case the draft of a basic rule set should not be bound to only one language.



Example of a multi lingual system

The rule-based transcription is organized quite multilingual in the usage. But it is already well-known from experience that each language is accessible by rules in different scale of a phonetising.

THE RULE FORMAT AND ITS TRACING BACK ON REGULAR RELATIONS

The rule format has initially a two-piece structure, which is adapted to the requirements of the regular relations. That means that to an input string of two types is always assigned an output string. For the output thereby is also possible the empty string. A notation would look like the following:

< Input string > < Output string >

An extension of this two-piece relation by a further structural component does not oppose in the principle. Therefore we make use of this possibility and insert as the third element a position indication. With this Information can be preset a range of a word, within whose a rule should be applicable.

It is to be mentioned that here under "Word" is to understand either a string of grapheme or a string of phone. A position indication has even also an internal structure, which can take two different forms. We have now a three-part format for the rules:

< Input string > < Output string > < Position >

The structure of input and output strings is presented now regarding their notation with two kinds of rules under fading out of a position indication.

CONTEXT RULES

With context rules it concerns around rules for grapheme or phone strings, which only can be used within a certain context on a certain 'Focus'. Observed from the top surface, a regular relation is a structure, which represents a linguistic rule.

A view of the internal structure of a regular relation induce us instead to talk about a regular relation as a quantity of condition transitions, which we can model with a transfer function.

SUBSTITUTION RULES

It concerns in this case the simple replacement rules. Their internal structure makes it possible to write out several rules parallel. The characteristic of the regularity is well readable with substitution rules in indicated format. This concerns regular relations of the type:

$$\mathtt{((I11 + \ldots + I1n) \circ . \ . \ . \circ (Im1 + \ldots + Imn),}$$
$$\mathtt{(O11 + \ldots + O1n) \circ . \ . \ . \circ (Om1 + \ldots + Omn))}$$

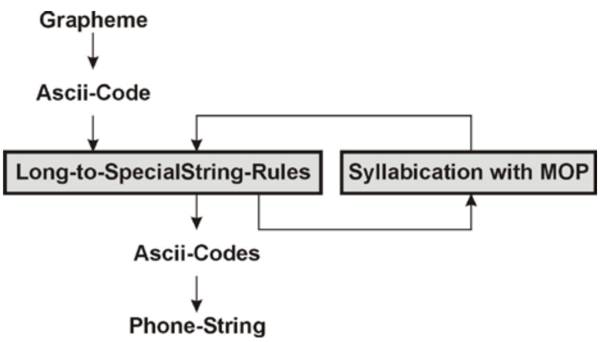
The variables Ijk and Ojk: j = 1… m : k = 1 ... n, stand for arbitrary strings from two different fundamental sets ΣI and ΣO.

For context rules are valid the same Rules as for the substitution rules with the difference that on the left side of the developing regular relation appears an additional focus and on the right side only one output string.

THE DATA TYPE AUTOMAT

The automat is understood as a vector of automat lines. An automat line again is a data type, which contains all necessary information for the description of the condition transitions.

Successively listed these lines result in a matrix-like table, in which a column is reserved either for grapheme or phone and the other one only for phone. Through a transfer into an internal representation, called "special string", the coding-technical conflict is resolved between the two groups of symbols of the grapheme and phone.



The structure of the rule based transcription

The theory of the regular relations and the finite State Transducer is transferred directly into the implementation. Each automat line has an entry for an input code, an output string, an indication concerning the current and the next condition. It is the initial situation for the concept of combination of the individual partial automats to a total automat and for the strategy for the efficient application of such.

ASSORTMENT OF THE AUTOMAT TABLE

The finished automats consist in practice of a table of several hundred thousand up to millions entries, depending on how many rules are needed for the setting up the automat.

Due to this large quantity of data it is inevitable to have funds to approach fast to the substantial information for the substantial processing steps. Remedy can be achieved through the advantage of the assortment. Instead of accomplishing the comparison for all data records, it is more favorable to use the internal code for the indexing of the table.

We make this more efficient operation not only with the application of the automats to the transcription, but also adopt with the construction of all possible condition transitions as ours own.

CONCLUSION

The regular expressions are suggested as means for the formulation of rules to the transcription of grapheme or phone strings in phone strings. They have the advantage that they are simply utilizable and fast easy to learn. By the introduction of variables to the terms of the regular expressions, easily comprehensible rules can be written, which remain still formal, but although approximate our used language.

From this basic idea results the consequence to avoid an overloading of the rule syntax and to treat separately the information from different structural levels of linguistics. This strategy adapt itself also the introduction of the XML notation.

As it is really created to build a heterogeneous knowledge base, from which individual parts then can be emerged as desired for the conversion of an algorithmic principle (like e.g. the maximum Onset Principle), in order to put them purposefully in a certain place.

To this concept is adapted the data structure "special string", which contains the necessary instructions for the precise realization of the transcription as bit string in coded form. Now for an appropriate evaluation of the procedure, the prepared rule sets by the linguists have to be applied to a suitable test case.

One may however quite expect that the results will fail more reliably in relation to the databased procedures, even if we can exclude that it concerns an ideal solution here.

REFERENCES

[1] Simsek, M.: Rule-based transcription of Text in international phonetic alphabet (IPA), Diplom thesis: Fakt. IV, TU-Berlin, Mai 2005.

[2] Mohri, M.: Weighted Finite-State Transducer Algorithms. An Overview ed. by Carlos Martin-Vide, Victor Mitrano, Gheorghe Paun / Formal Languages and Applications vol. 148 VIII/Springer Verlag Berlin, 2004.

[3] Lehmann, C. Phonetik und Phonolgie / sprachwissenschaftliche Fak. der Uni Erfurt, 2004.

[4] Jurafsky, D.; Martin, J. H. / Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition / Prentice Hall / New Jersey, 2000.