Design and Implementation of the Slovenian Phonetic and Morphology Lexicons for the Use in Spoken Language Applications

Matej Rojc, Zdravko Kačič,
Darinka Verdonik

Faculty of Electrical Engineering and Computer Science, University of Maribor
Smetanova 17, 2000 Maribor, Slovenia
{matej.rojc, kacic}@uni-mb.si

Phonetic and Morphology Lexicons that can be used in Spoken Language Applications are costly and time-consuming to build. This paper reports on a project aiming at the semi-automatic development of large phonetic (*SIflex*) and morphology (*SImlex*) lexicons for Slovenian language. The main goal of the project is to build the phonetic and morphology lexicon for Slovenian language that will be used within the framework of various applications in speech processing (e.g. speech synthesis and recognition), natural language processing (e.g. spell checking) and for studying and assessing automatic grapheme-to-phoneme transcription. In automatic speech recognition one of the major problem is extremely high variability of pronunciations. One part of this variability can be taken into account through a training of the acoustic-phonetic units from a large amount of data. Another part of variability must be modeled in the lexicon as pronunciation variants. In the case of text-to-speech systems it is also very usable to be able to detect homographs and choose the correct pronunciation according to the context information. All this was our motivation for developing both lexicons for Slovenian language. Currently the created phonetic lexicon (*SIflex*) contains more than 130.000 items, whereas the morphology lexicon (*SImlex*) consists of approximately 600.000 inflected forms, including information on the orthography, pronunciation, stress and morphosyntactic features, as defined in the framework of the Multext project.

1. Introduction

Development of real models of human language that support research and technology development in language related fields require a lot of linguistic data: lexicons of thousands of words. In case of inflectional languages such lexicons must be up to ten times larger, to be able to achieve the same coverage, as in the case of e.g. English language. A lot of Slovenian root forms can result in up-to 200 different inflectional forms. The use of such resources can therefore represent substantial computational load. A given spoken language system, which uses fully inflected word forms, performs much worse with highly inflected languages (e.g. Slovenian) than with non or purely inflected languages (e.g. English), where the lexicons used can be much smaller.

2. System architecture

In figure 1 the system for lexicons' development is shown. This figure shows the system architecture, which consists of basically two levels. Firstly the data preparation step is performed, followed by the lexicons' build-up level. The lexicons' build-up level consists of two modules: a rule-based linguistic module and an automatic grapheme-to-phoneme conversion module. The user graphic interface links all the modules together. A more detailed description of the system modules is given later. The architecture of the system is modular and multilingually oriented. Appropriate system modules have to be adopted for the use of presented platform in other languages. The tokenization module is capable of multilingual text processing (Rojc,1999), and the statistical part of the grapheme-to-phoneme conversion module can also be considered as multilingual, since it uses a data-driven approach based on neural networks. The graphic user interface currently supports Slovenian language, but can be adapted for other languages as well. All modules are written in C++ program language, except the graphic interface, which is written in Java language using Visual J++. The whole system runs on all Windows platforms – Windows 95/98, Windows NT/2000/XP.

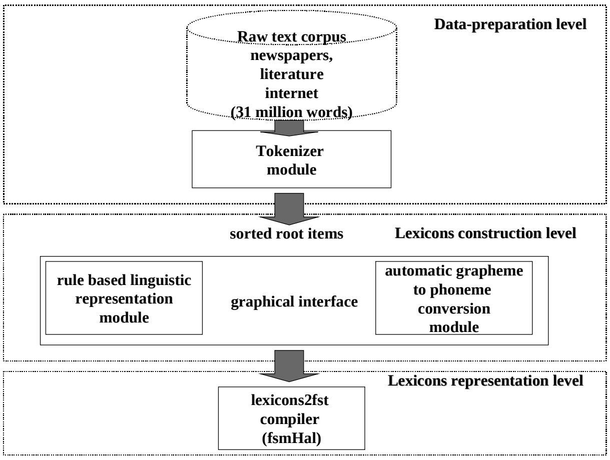

Data-preparation level
Raw text corpus
newspapers,
literature
internet
(31 million words)
Tokenizer module
sorted root items
Lexicons construction level
rule based linguistic representation module
graphical interface
automatic grapheme to phoneme conversion module
Lexicons representation level
lexicons2fst compiler (fsmHal)

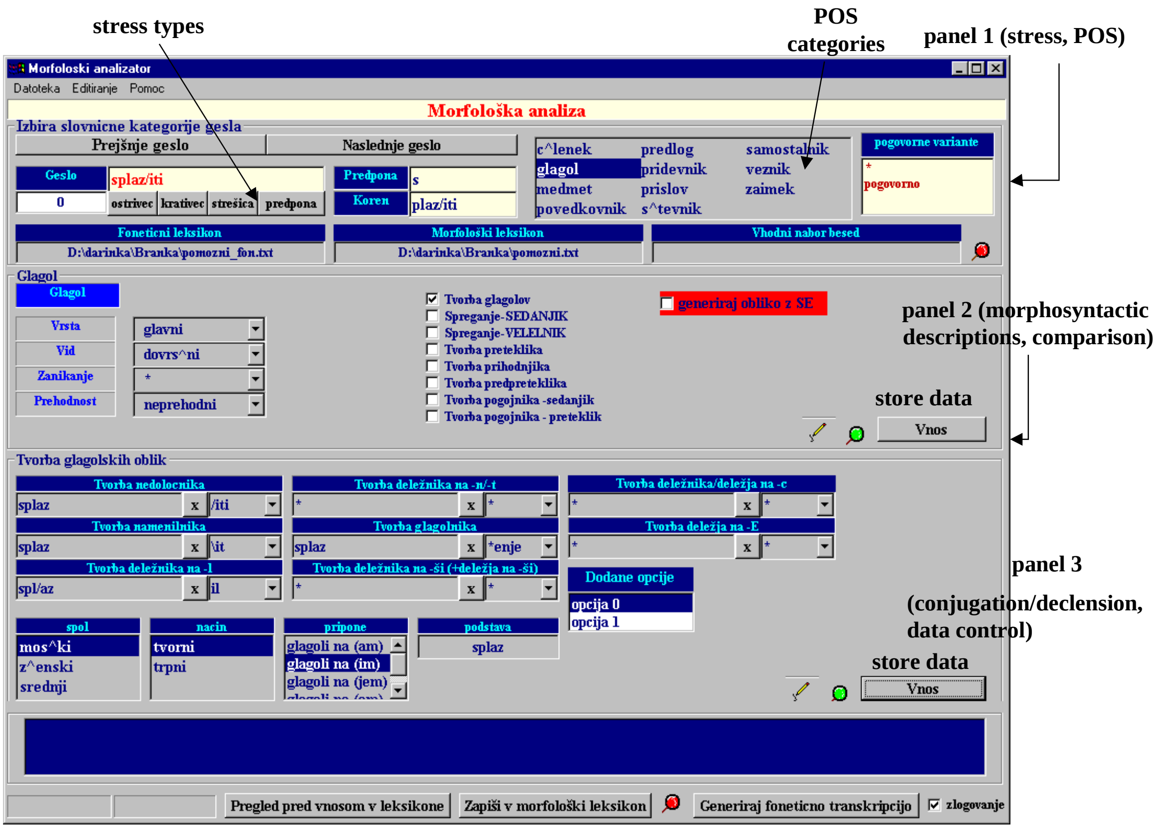
Figure 1. System architecture for building morphological and phonetic lexicons.

2.1. Tokenization and word selection process

Some text pre-processing on the obtained text corpus has to be done before using platform for building morphology and phonetic lexicons (e.g. for Slovenian language). These algorithms have to be highly flexible and robust according to the general, unrestricted nature of the text. The input text corpus (raw ASCII text) is fed into a tokenizer module in Fig. 1 (finite-state machine) (Rojc,1999), which emits hypotheses about tokens, and segments the input text into words.

The tokenizer module is organised on a multilevel basis. At the lowest level the lexical scanner separates the input text into tokens. Some tokens may not be in a canonical form appropriate for building up morphology and phonetic lexicons. In this case the text normalization processing level breaks such tokens into their constituent words.

All tokens such as date, hour, cardinal, and ordinal numbers are expanded into corresponding word forms during the tokenization process in the tokenizer module ('expanding text processing level'). The obtained words are sorted and a word frequency is assigned to each one. A final list of items was defined with this procedure, using the 30.000 most frequent words in the input corpus (root forms).



Geslo: splaz/iti
0
Predpona: s
Koren: plaz/iti
pogovorne variante: * pogovorno
Foneticni leksikon: D:\darinka\Branka\pomozni_fon.txt
Morfološki leksikon: D:\darinka\Branka\pomozni.txt
Vhodni nabor besed:

Glagol
Vrsta: glavni
Vid: dovrs^ni
Zanikanje: *
Prehodnost: neprehodni

☑ Tvorba glagolov
☐ Spreganje-SEDANJIK
☐ Spreganje-VELELNIK
☐ Tvorba preteklika
☐ Tvorba prihodnjika
☐ Tvorba predpreteklika
☐ Tvorba pogojnika -sedanjik
☐ Tvorba pogojnika - preteklik
☐ generiraj obliko z SE

Tvorba nedolocnika: splaz x /iti
Tvorba deležnika na -n/-t: * x *
Tvorba deležnika/deležja na -c: * x *
Tvorba namenilnika: splaz x \it
Tvorba glagolnika: splaz x *enje
Tvorba deležja na -E: * x *
Tvorba deležnika na -l: spl/az x il
Tvorba deležnika na -ši (+deležja na -ši): * x *
Dodane opcije: opcija 0, opcija 1
spol: mos^ki, z^enski, srednji
nacin: tvorni, trpni
pripone: glagoli na (am), glagoli na (im), glagoli na (jem)
podstava: splaz
☑ zlogovanje

Figure 2: Graphic interface of the system for building morphologic and phonetic lexicons

2.2. Rule based linguistic representation module

Figure 2 shows the graphic interface of the system (lexicons construction level). This interface visualizes all the information generated by the rule-based linguistic module (e.g. adjective). When using it the expert is able to perform editing, correction, and verification actions. The morphology analysis represents the main part of the system. The graphic interface consists of three panels. The first panel is fixed and is intended as a starting point for any analysis the expert has to perform. The linguistic expert has to load the already created phonetic and morphology lexicons or create a new one. As input a list of all items is also loaded, to which phonetic and morphology information will be assigned. Then the expert manually chooses the item from the list or just types the new one. If the item is not in root form, it must be corrected. When the preprocessing of raw text is performed, all the items on the list are already in the root form and alphabetically ordered. Next, the expert verifies the type and position of the stress and marks a suffix in the item if it exists. The syntactic category for the corresponding item (part-of-speech) is further chosen: noun, verb, adjective, number, pronoun, adverb, conjunction, interjection, article, proposition and predicative.

This action opens the appropriate second and third panel for corresponding parts-of-speech. According to this selection, the rule-based linguistic representation component generates and places all attributes and values into their respective fields. The linguistic expert than verifies all the values and corrects them if needed. The data control window serves for verification of the generated data. The expert can define the comparative forms of adjectives, adverbs and the data for lemmas using the second panel. The third panel is used for the declension or conjugative forms of lemmas.

Currently this component works for the Slovenian language only, but can also be adapted to other languages, integrating the appropriate linguistic rules. The Slovenian language is like other Slavic languages inflectional language and the linguistic representation of a word depends on complex contextual factors. Most general linguistic rules were integrated into this component. Since there are quite a lot of exceptions in the Slovenian language, a linguistic expert's verification has to be done after the automatic generation of all forms for the current item's linguistic category (POS – part of speech). All the obtained results are added to the existing morphology lexicon in the prescribed format, after the linguistic expert verifies them. Morphology lexicon SImlex includes information on the stress and morphosyntactic features, as defined in the framework of the Multext project (Multext, 1996).

Some basic actions that have to be performed for specific part-of-speech categorization are defined below:

- adjective: the comparison is automatically performed and a conjugation/declension panel has to be activated,
- nouns: appropriate declension must be chosen and a conjugation/ declension panel activated,
- verbs: several panels for building various verb forms have to be activated and verified after automatic generation (infinitive, supine, participle, verb conjugation etc.),
- number: the expert determines gender, number etc. for root form and activates a conjugation/declension panel,
- pronoun: its type, gender, person, number and case are determined for root form, and a conjugation/ declension panel also activated,
- adverb: comparison is automatically performed and the type chosen,
- conjunction: the type is determined,
- interjection: the type is determined,
- article: the type is determined,
- predicative: the type and case are determined.

When the expert verifies the content of the second panel, the information is saved into the memory by pushing the 'store data' button. The expert then moves to the third panel (conjugation/declension), where automatically generated conjugation/declension forms of the root item are performed (adjective, nouns, verbs, and numbers). In the Slovenian language there are many exceptions, which cannot always be correctly interpreted by the rule-based linguistic representation component. Sometimes the stress changes the position and type in the word during the conjugation/declension process. Manual corrections by the expert are needed since this is very hard to predict using rules.

2.3. Automatic grapheme-to-phoneme conversion module

The words in their grapheme representation have to be mapped onto their phonetic representation i.e. into their pronunciation for speech recognition and text-to-speech synthesis. The availability of pronunciation lexicons in their canonical form (a single phonemic representation for each word) is very important in order to be able to build grapheme-to-phoneme models. In some languages this form can be derived from the grapheme form by a set of pronunciation rules, but for many languages (e.g. for the Slovenian language) this relationship is complex and is nowadays usually handled by manually produced pronunciation lexicons.

In many languages the number of phones in its pronunciation and the number of letters in an orthographic transcription of a word are not a one to one match. Letters can usually map to zero, one or two phones. When letters in some contexts correspond to no phone, they are marked with an empty symbol (_epsilon_). The alignment task actually becomes introduction of epsilons into the phonemic representation so that it matches the length of the grapheme representation. The explicit listing of which phones (or multi-phones) each letter in the alphabet may correspond to, irrespective of the context, is defined first before string alignment. This task can be done in an interactive process over the training set from the phonetic lexicon. New correspondence is added to the list of allowable mappings during this process.

In the list of allowable mappings, the vowels have a much longer list of potential phones. The input of the used algorithm is the list of allowable mappings and the input lexicon. The probability that the grapheme G is matching with one phoneme P is estimated during processing, and the DTW is used for introducing epsilons at positions that maximize the probability of the word's alignment path. Once the dictionary is aligned, the association probabilities can be computed again, and so on until convergence.

The grapheme-to-phoneme conversion is performed in two steps. In the first, stress marks are inserted. In the second, the graphemes are converted into phonemes and the syllable breaks inserted in the phoneme string. The neural networks (NNs for stress determination and grapheme-to-phoneme conversion) are constantly learned off-line during the lexicons' development and then integrated into the grapheme-to-phoneme conversion module to increase its performance. A multilayer perceptron (MLP) feedforward network with one hidden layer was used for neural networks. A backpropagation algorithm was chosen as the learning algorithm for both networks (Zell,1994, Hain,1999).

Marking syllables is a very important and necessary operation in text-to-speech synthesis systems since in many languages the pronunciation of phonemes is a function of their location in the syllable, relative to the syllable boundaries.

Phoneme location in the syllable also has an important role for the duration of the phone and represents significant information for any module that predicts segmental duration in the text-to-speech synthesis system.

2.3.1. Syllable structure of the Slovenian language

The Slovenian language allows complex consonant clusters in the onset and coda of the syllables. Some examples are given in table 2. The classes of Slovenian phones are presented in table 1. Statistical results showed that the Slovenian language allows up to 4 consonants in the onset and up to 4 consonants in the coda of the syllable. Determining the syllable boundary correctly is important because the pronunciation of most phonemes is a function of their position in the syllable. Syllable boundaries also influence phoneme duration.

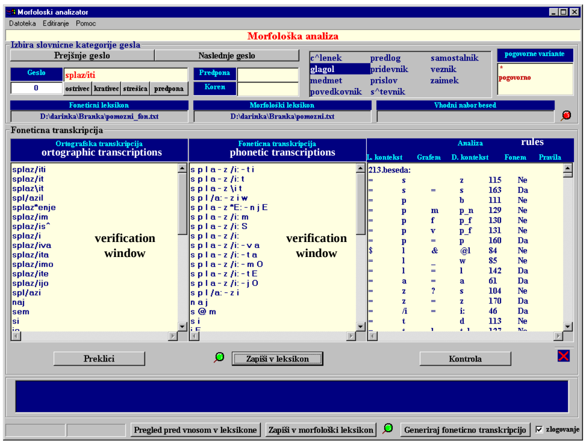
class	Description	Slovenian phones
P	unvoiced stops	p t k
B	voiced stops	b d g
S	unvoiced fricatives	f s S x
Z	voiced fricatives	z Z
N	nasals	m n
L	liquids, glides	l r v j
V	vowels	i: e: E: a: O: o: u: i E a O u @

Table 1: Classes of Slovenian phones (Sampa,1998).

The syllabification process is implemented as a weighted finite-state transducer (WFST) that is constructed from a list of syllables (Kiraz,1998). The weights of the finite-state transducer were determined using frequencies of onset, nucleus, and coda types obtained from the statistical processing of training data (syllables). They play a significant role in obtaining the correct syllabification, especially in the case of consonant clusters.

Onsets		Codas	
class	clusters	class	clusters
PLL	klj	LSP	jst
SPL	skr	LSPN	jstn
SNL	Snj	SPP	stk
LLL	vrv	BLL	brv
ZBL	zbr	BLP	brt

Table 2: Slovenian language allows up to 4 consonants in the onset of the syllable and 4 consonants in the coda of the syllable.



Geslo: splaz/iti

0

c^lenek, predlog, samostalnik, glagol, pridevnik, veznik, medmet, prislov, zaimek, povedkovnik, s^tevnik

*
pogovorno

Foneticni leksikon: D:\darinka\Branka\pomozni_fon.txt

Morfološki leksikon: D:\darinka\Branka\pomozni.txt

```
splaz/iti      s p l a - z /i: - t i
splaz/it       s p l a - z /i: t
splaz\it       s p l a - z \i t
spl/azil       s p l /a: - z i w
splaz*enje     s p l a - z *E: - n j E
splaz/im       s p l a - z /i: m
splaz/is^      s p l a - z /i: S
splaz/i        s p l a - z /i:
splaz/iva      s p l a - z /i: - v a
splaz/ita      s p l a - z /i: - t a
splaz/imo      s p l a - z /i: - m O
splaz/ite      s p l a - z /i: - t E
splaz/ijo      s p l a - z /i: - j O
spl/azi        s p l /a: - z i
naj            n a j
sem            s @ m
si             s i
```

```
L. kontekst  Grafem  D. kontekst  Fonem  Pravila
213.beseda:
=            s                    z      115  Ne
=            s       =            s      163  Da
=            p                    b      111  Ne
=            p       m            p_n    129  Ne
=            p       f            p_f    130  Ne
=            p       v            p_f    131  Ne
=            p       =            p      160  Da
$            l       &            @l     84   Ne
=            l       _            w      85   Ne
=            l       =            l      142  Da
=            a       =            a      61   Da
=            z       ?            s      104  Ne
=            z       =            z      170  Da
=            /i      =            i:     46   Da
=            t                    d      113  Ne
```

Figure 5*: Grapheme-to-phoneme conversion*

To verify automatically generated phonetic transcription, the expert has to open the window, which visualizes generated phonetic transcription, done with an automatic grapheme-to-phoneme conversion module. This window is shown on Figure 5. The user is actually able to switch between using rule-based or data-driven approaches (using a neural network). Only different orthographic words are sent to the grapheme-to-phoneme transcription module since a lot of inflectional forms for corresponding items are the same.

The grapheme-to-phoneme transcription module receives as an input only non-duplicated words and returns the corresponding phonetic transcriptions with syllable break marks '-' and stress marks. The linguistic expert verifies the results and writes everything into the phonetic lexicon. The control window in Figure 5 can be used for inspection (which rules were applied) to identify problems in the case of errors in the obtained phonetic transcription (when the rule-based approach is used).

3. Project status

The project of building Slovenian phonetic (*SIflex*) and morphology lexicons (*SImlex*) started in 1999. Currently six linguistic experts are working on this project, using the presented system. The system described was evaluated as a very efficient help for the expert. The linguistic experts found it very easy to use, accurate enough in automatic generation of linguistic descriptions for items and also in grapheme to phoneme transcriptions. In case of errors, they can be corrected fast and easy without extensive typing. To avoid undesired errors, the analysis and correction can be performed mostly by using mouse. The experts are able to verify in average 30 root items per hour (phonetic and morphologic lexicons). The presented platform is constantly improved in order to speed up the lexicon construction process. Since the Slovenian language is a very inflectional language, on average 30 inflectional forms per root item are generated (during analysis of verbs up to 200 inflectional forms can be generated). The phonetic lexicon is smaller than the morphologic lexicon, since a lot of duplicated inflectional forms are obtained during conjugation/declension. The SImlex morphologic lexicon currently contains more than 600.000 items (root items plus corresponding inflectional forms) and SIflex phonetic lexicon about 130.000 items.

4. Compilation of large scale lexicons into finite-state transducers

To be able to use large-scale lexicons in spoken dialogue applications, we need to use the efficient way for their representation. The methods used in the compilation of both lexicons into finite-state transducers (FST) assume that lexicons are given as large lists of strings and not as a set of rules as for instance in (Mohri,1995). Such representation is time and space optimal and very appropriate for the use in spoken language applications, especially in the case of inflectional languages.

In the compilation process a large set of proprietary programs (*fsmHal*) written in C++ were used that perform efficiently many operations on finite-state transducers and finite-state automata including determinization, minimization, union, intersection, compaction, prefixation, local extension and others.

The following algorithms were used during the construction of corresponding finite-state transducers: union, determinization, prefix computation, and classical minimization algorithms of finite-state automaton (Aho, 1974; Watson, 1995). Prefix computation algorithm (Mohri,1995) was used before minimization algorithm.

	FST_1	FST_2
Number of states	69.498	90.613
Number of transitions	90.801	130.839
Size of bin file	252 kB	662 kB

Table 6: The final finite-state transducers representing Slovenian phonetic (FST_1) (60.000 items) and Slovenian morphology lexicon (FST_2) (40.000 items).

The representation using finite-state transducers was performed for the *SIflex* and *SImlex* Slovenian lexicons. The starting size for SIflex was 1.8 MB (60.000 items) and 1.4 MB for SImlex (40.000 items). The final size achieved using the presented algorithms was 352 kB for SIflex and 662 kB for SImlex. Representation of large lexicons using finite-state transducers is mainly motivated by considerations of space and time efficiency. For both lexicons a great reduction in size and optimal access time was achieved. Using such representation the look-up time is optimal, since it depends only on the length of the input word and not on the size of the lexicon.

5. Conclusion

It is planned to use the Slovenian phonetic lexicon for research work in the field of automatic continuous speech recognition for the Slovenian language. Using both lexicons (*SIflex* and *SImlex*) it is also possible to automatically detect homographs contained in the lexicons. This information is very useful in the text-to-speech synthesis systems for Slovenian language, since the input text can contain up to 30% of homographs. Both phonetic and morphology lexicons will be used for Slovenian text-to-speech synthesis. Minimal memory usage and fast look-up times are desired, when using lexicons in run-time systems. Lexicons can be very extensive, therefore it is very important that their representation is optimal. As shown in this paper, the representation of lexicons using finite-state transducers fulfils both requirements. They provide fast look-up time, double side look-up, and compactness.

6. References

Mehryar Mohri., (1995) *On Some Applications of Finite-State Automata Theory to Natural Language Processing,* Natural Language Engineering 1, Cambridge University Press.

George Anton Kiraz and Bernd Möbius., Multilingual Syllabification Using Weighted Finite-state Transducers, Bell Labs – Lucent Technologies (1998).

SAMPA for Slovenian, (1998) http://www.phon.ucl.ac.uk/home/sampa/slovenian.html

MULTEXT project lexical specifications, (1996) http://www.lpl.univaix.fr/projects/multext/LEX/LEX.Specifications.html

Matej Rojc, Janez Stergar, Ralph Wilhelm, Horst-Udo Hain, Martin Holzapfel, Bogomir Horvat, (1999) A Multilingual text processing Engine for Text-To-Speech Synthesis system, Proceedings EUROSPEECH 1999, Budapest, pp. 2107-2110

Bruce William Watson, (1995) *Taxonomies and Toolkits of Regular Language Algorithms*, PhD Thesis, Eindhoven University of Technology and Computing Science.

Horst-Udo Hain, (1999) Automation of the training procedure for neural networks performing multi-lingual grapheme to phoneme conversion, Proceedings EUROSPEECH 1999, Budapest, pp. 2087-2090

Zell, A.(1994). Simulation Neuronaler Netze. Bonn, Paris; Reading, Mass., Addison-Wesley

Aho, Alfred V., John E. Hopcroft, and Jeffrey D. Ullman, (1974) *The design and analysis of computer algorithms*. Addison Wesley: Reading, MA.