Implementation for discovery:
A bipartite lexicon to support morphological and syntactic analysis

Emily M. Bender and Jeff Good
ebender@u.washington.edu and good@eva.mpg.de
University of Washington and MPI EVA
CLS 41, April 8, 2005

Outline of talk

- Introduce Montage project
- Discuss the general Montage approach to morphology
- Outline current implementation of the approach using the LKB and XFST
- Discuss some outstanding issues in the implementation design

Overview: Montage

- Suite of tools to assist in the documentation of underdescribed languages (Bender et al. 2004)
- Focus on grammar (especially morphology and morphosyntax)
- Integrate with other initiatives building tools for transcribed texts and lexicons (e.g., ELAN, FIELD, AGTK)

Overview: Montage

- Overarching goal: Allow the "ordinary working linguist" (or OWL) to make use of sophisticated grammar engineering tools without being grammar engineers themselves
- This talk: The Montage model under development for morphological analysis, with a focus on morphophonological analysis

Terminology

- Morphophonology
  - Morphotactics (e.g., position classes)
  - Phonological/morphophonological rules
  - Mapping to abstract morphemes
- Morphosyntax
  - Syntactic-semantic representations built from analysis of strings of abstract morphemes

Morphology in Montage

- Three possible models
  - Morphophonology in morphosyntax (see, e.g., Baker's (1988) notion of *incorporation*)
  - Morphosyntax in morphophonology (perhaps, Beesley & Karttunen's (2003: 343–349) analysis of Arabic case using flag diacritics)

Morphology in Montage

- Three possible models (contd.)
  - Morphophonology independent from morphosyntax (see Woodbury (1996) for one articulation)
  - This is the model adopted by Montage

Morphology in Montage

- Why morphophonology independent from morphosyntax?
  - Some languages, like Athabaskan languages, are traditionally analyzed as such, and we need to support such analyses
  - Gives the documentary linguist flexibility in dealing with partially analyzed data

Morphology in Montage

- Separating morphophonology and morphosyntax also fits with a core philosophical principle of Montage: Use existing tools wherever possible
- XFST (Beesley and Karttunen 2004)
- LKB (Copestake 2002)
- No one tool has the combined functionality of these two existing tools

Morphology in Montage

- Some problems with morphophonology within morphosyntax
  - Hard to “reuse” morphophonological analyses
  - Particularly awkward for strictly phonological effects
  - The morphophonology is more efficient if it can be pushed into one (finite-state) machine

Morphology in Montage

- Some problems with morphosyntax within morphophonology
- This could be exemplified by a representation like:
  boys → 'boy.[NUM plural]'
- Such representations won't work well for "hard" cases like causatives or passives

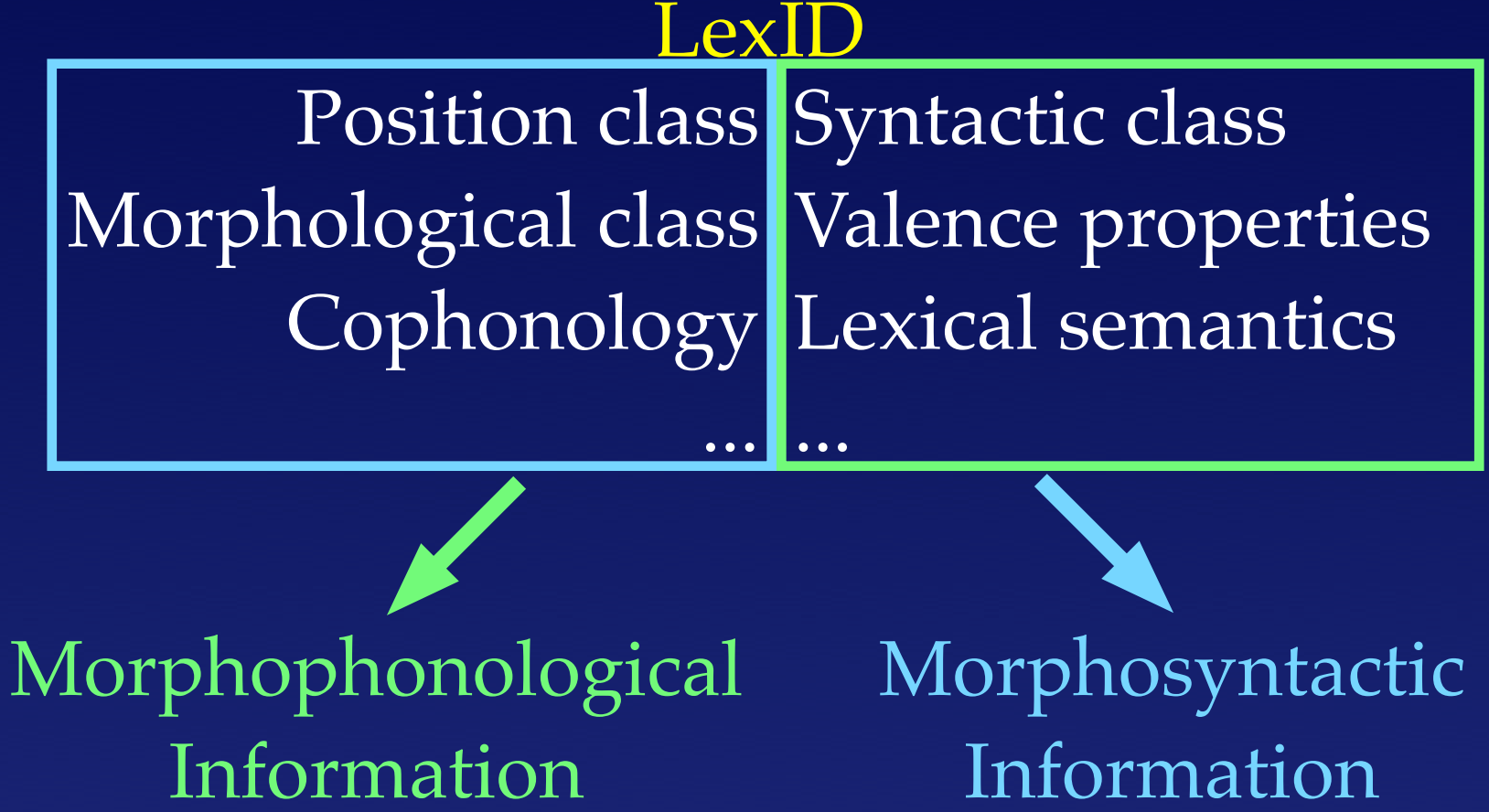
The Interface

- Separated morphophonology and morphosyntax need to be interfaced in some way
- This is done in Montage through the use of a bipartite lexical database
- Critically, this interface means that the morphophonological component is not completely a “black box”

Lexical Database

- Each lexical entry is associated with
  - A Lexical ID
  - Morphophonological information
  - Morphosyntactic information
- The sort of information in each part of the entry could be customized by the linguist, ideally assisted by lexical templates

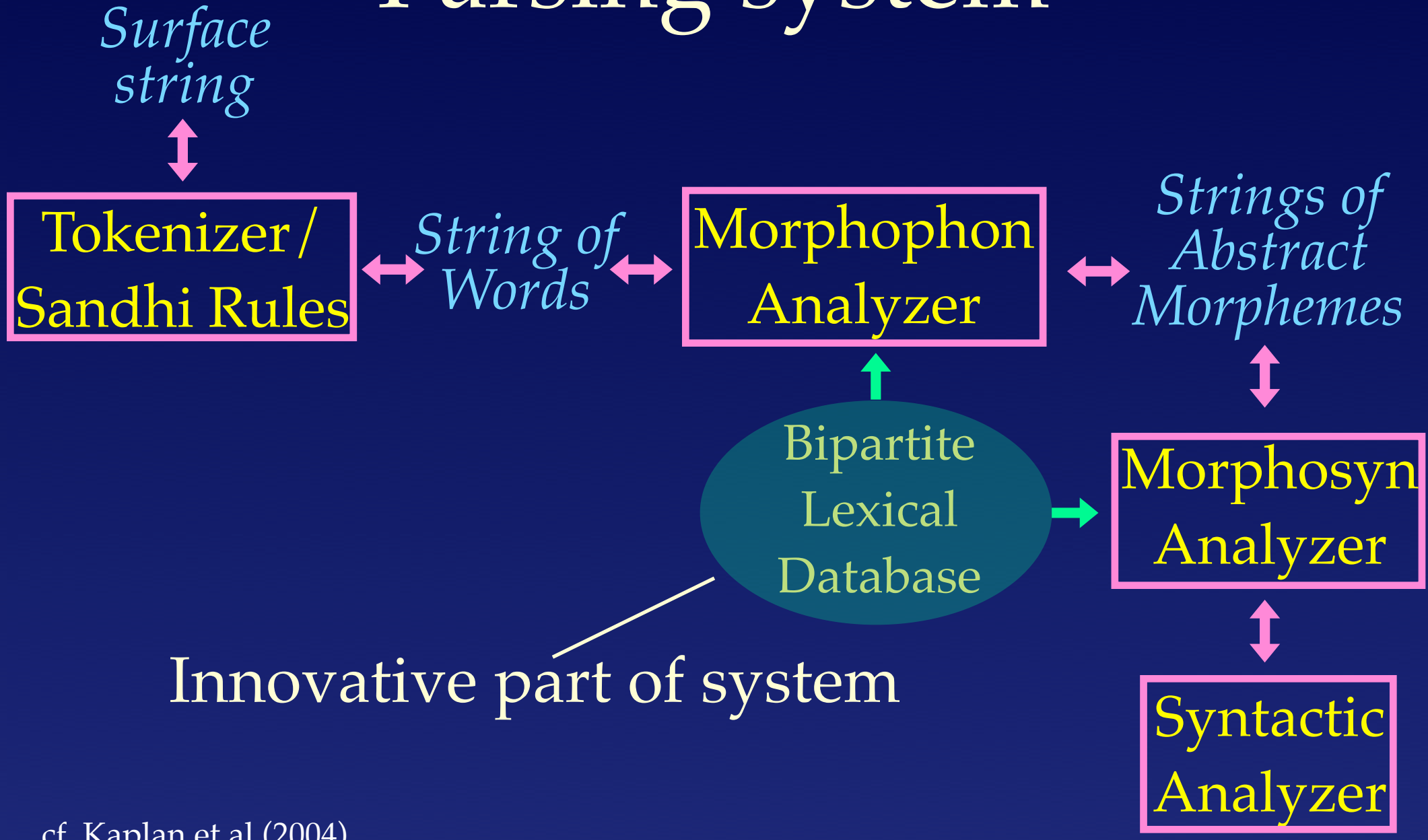
Lexical Database



Lexical Database

- In the present implementation, the lexicon, in fact, has a third component
- It has proven practical to keep lexicographic information (e.g., citation form and gloss) separate from more strictly grammatical information

Parsing system



cf. Kaplan et al (2004),
Siegel and Bender (2002)

Implementation

- The current implementation of the morphophonological analyzer makes use of several resources in addition to the lexicon
- The most complex of these is a series of morphophonological rule definitions which can be classified according to their type (e.g., verbal, nominal, “general”)

Implementation

- Two additional resources are
  - A set of character class definitions allowing the use of natural classes of segments in morphophonological rules
  - A set of position class definitions for specifying the properties of relevant morphological "position classes" (e.g., whether the position is optional or obligatory)

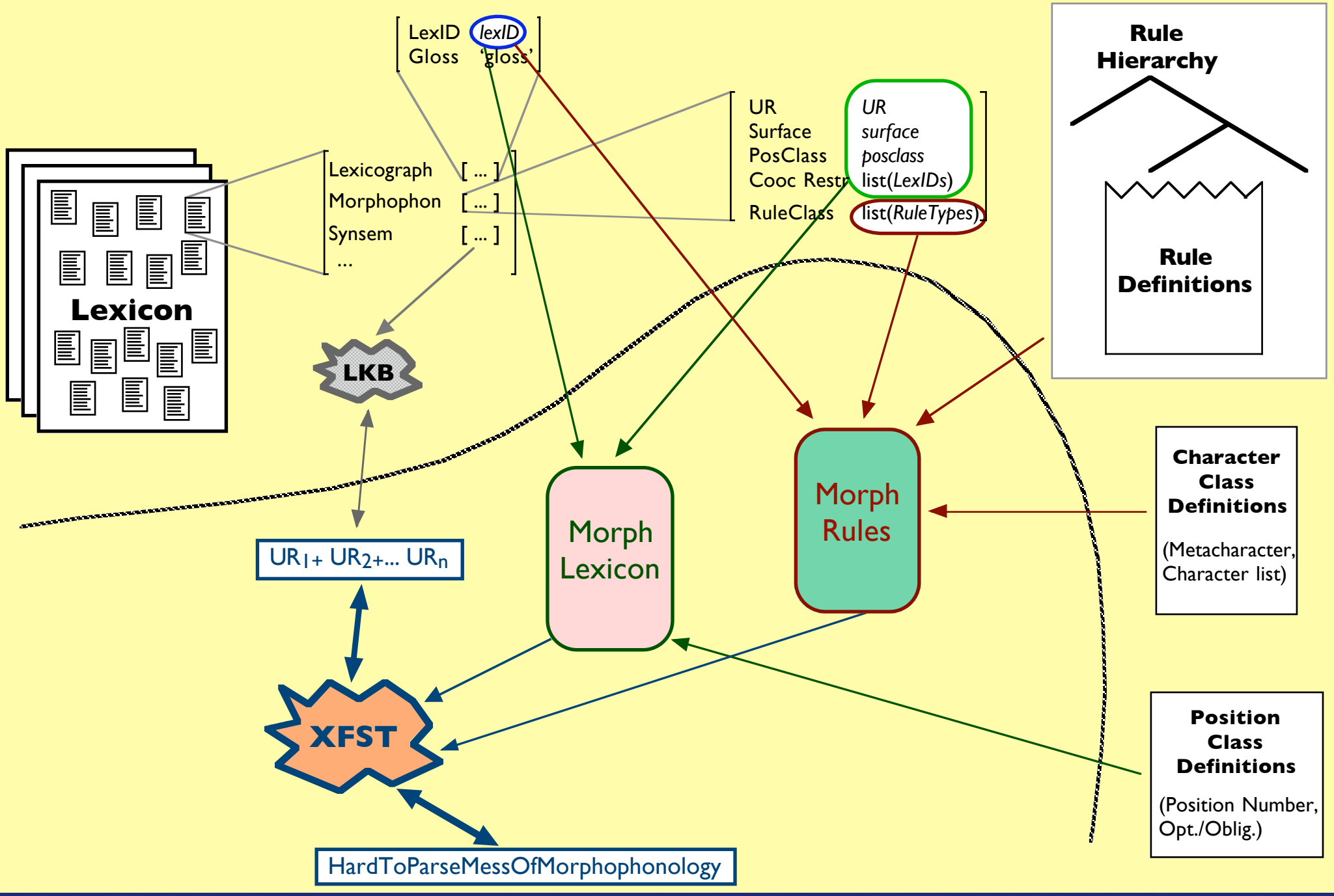
Implementation

- Morphophonological features used in the lexicon at present
  - “Underlying representation”
  - “Surface representation”
  - Position Class
  - Coocurrence restrictions
  - Rules associated with

Implementation

- Important technical details of the implementation
  - Morphosyntax/semantics is handled using the LKB (Copestake 2002) based on grammars built using the Grammar Matrix (Bender et al. 2002)
  - Morphophonological parsing is handled through XFST

Implementation



Implementation

- Important features of the system
  - Bidirectionality—parsing and generation
  - Bipartite lexicon allows incremental modification of analyses of morphosyntax and morphophonology

Future work

- Development of a user interface of a sort an OWL would be comfortable with
- Establishing a general system for dealing with rule ordering
- Support for the statement of "construction-level" morphophonological generalizations (for example, stem/word minimality constraints)

Future work

- An area of research right now is how big of a role the morphosyntax should have in filtering out morphophonological parses
- For example, when should coocurrence restrictions be handled by the morphophonology versus the morphosyntax?

References

Baker, Mark. 1988. *Incorporation: A theory of grammatical function changing*. Chicago: University of Chicago.

Beesley, Kenneth R. and Lauri Karttunen. 2003. *Finite State Morphology*. Stanford: CSLI.

Bender, Emily M., Dan Flickinger, Jeff Good and Ivan A. Sag. 2004. Montage: Leveraging advances in grammar engineering, linguistic ontologies, and mark-up for the documentation of underdescribed languages. *Proceedings of the Workshop on First Steps for the Documentation of Minority Languages: Computational Linguistic Tools for Morphology, Lexicon and Corpus Compilation, LREC 2004*. Lisbon, Portugal.

Bender, Emily M., Dan Flickinger and Stephan Oepen. 2002. The Grammar Matrix: An Open-source starter-Kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*. Taipei, Taiwan. pp. 8-14.

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. Stanford: CSLI.

Rice, Keren. 1989. *A grammar of Slave*. Berlin: Mouton.

Siegel, Melanie and Emily M. Bender. 2002. Efficient deep processing of Japanese. In *Proceedings of the 3rd workshop on Asian language resources and international standardization at the 19th International Conference on Computational Linguistics*. Taipei, Taiwan.

Woodbury, Anthony C. On restricting the role of morphology in Autolexical syntax. In E. Schiller et al. (eds), *Autolexical theory: Ideas and methods*. Berlin: Mouton. pp. 318–363.

Abbreviations

AGTK	Annotation Graph Toolkit.
	http://www.ldc.upenn.edu/Projects/AG/
ELAN	EUDICO Linguistic Annotator.
	http://www.mpi.nl/tools/elan.html
FIELD	Field Input Environment for Linguistic Data.
	http://emeld.org/tools/fieldinput.cfm
Grammar Matrix	Precision Grammar Starter Kit.
	http://www.delph-in.net/matrix/
LKB	LKB Grammar Development Environment.
	http://ww.delph-in.net/lkb/
XFST	Xerox Finite State Transducer.
	http://www.fsmbook.com/

Acknowledgments

Thanks to Anya Dormer for her work on an XFST implementation of Slave morphophonology and to Duane Blanchard, Scott Drellishak, Ann Gaponoff, David Goss-Grubbs, Jeremy Kahn, Mike Maxwell, Bill McNeill, Matty Noble, and Laurie Poulson for helpful discussion