

# Treeex: Modular NLP Framework

**Martin Popel**

ÚFAL (Institute of Formal and Applied Linguistics)  
Charles University in Prague



September 2011, Malá Skála

# Outline

- Motivation, Treex vs. TectoMT
- Treex architecture
- Treex internals
- Future plans
- Conclusion and examples

# Motivation

## Goals of Treex

- elegant integration of in-house and third-party NLP tools
- modularity, reusability, cooperation
- ability to easily modify and add code in a full-fledged programming language (Perl)

# Treex vs. TectoMT



2005 (Zdeněk Žabokrtský)

**NLP framework**  
*TectoMT*

**MT system**  
*TectoMT*

lemmatization

tagging

parsing

# Treex vs. TectoMT



2005

...

2011

NLP framework  
*TectoMT*

multi-purpose  
NLP framework  
*Treex*

MT system  
*TectoMT*

MT system  
*TectoMT*

lemmatization

lemmatization

tagging

tagging

parsing

parsing

coreference

PEDT preprocessing

CzEng analysis

treebank conversions

named entity r.

alignment (word,tree)

SMT preproc.

etc.

# Treex vs. TectoMT



2005

...

2011

NLP framework  
*TectoMT*

multi-purpose  
NLP framework  
*Treex*

MT system  
*TectoMT*

lemmatization  
tagging  
parsing

MT system  
*TectoMT*

lemmatization  
tagging  
parsing

coreference  
CzEng analysis  
named entity r.  
SMT preproc.

PEDT preprocessing  
treebank conversions  
alignment (word,tree)  
etc.

Now not only  
tectogrammatrics  
and not only MT  
→ renamed



# Treex vs. TectoMT



2005

...

2011

NLP framework  
*TectoMT*

multi-purpose  
NLP framework  
*Treex*

MT system  
*TectoMT*

lemmatization  
tagging  
parsing

MT system  
*TectoMT*

lemmatization  
tagging  
parsing

coreference  
CzEng analysis  
named entity r.  
SMT preproc.

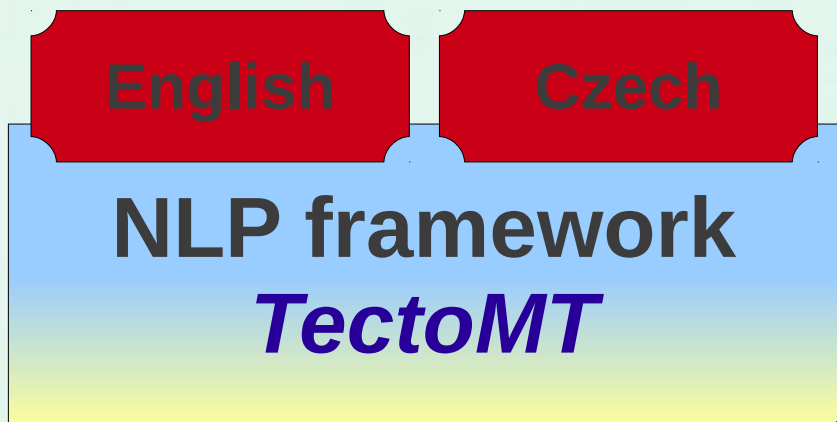
PEDT preprocessing  
treebank conversions  
alignment (word,tree)  
etc.

redesigned and  
reimplemented  
➔ easier to use  
➔ more flexible

# Treeex vs. TectoMT

2005

...

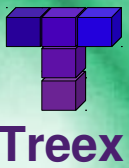


- redesigned and reimplemented
- ➔ easier to use
  - ➔ more flexible
  - ➔ more langs

\*) Most of the listed languages are only drafts of analysis made by students, not converted to Treeex yet. The entire risk as to the quality and performance of the program is with you.



# Treex vs. TectoMT



2005

English

English

Czech

**Special offer**  
**Call now and get**  
**one extra Treex**  
**for free**

Hindi

Network

Esperanto

French

M...  
**Tecto**

German

Arabic

Coherence

PEBT preprocessing

Vietnamese

Hindi

named entity r...

alignment (word tree)

Urdu

Finish

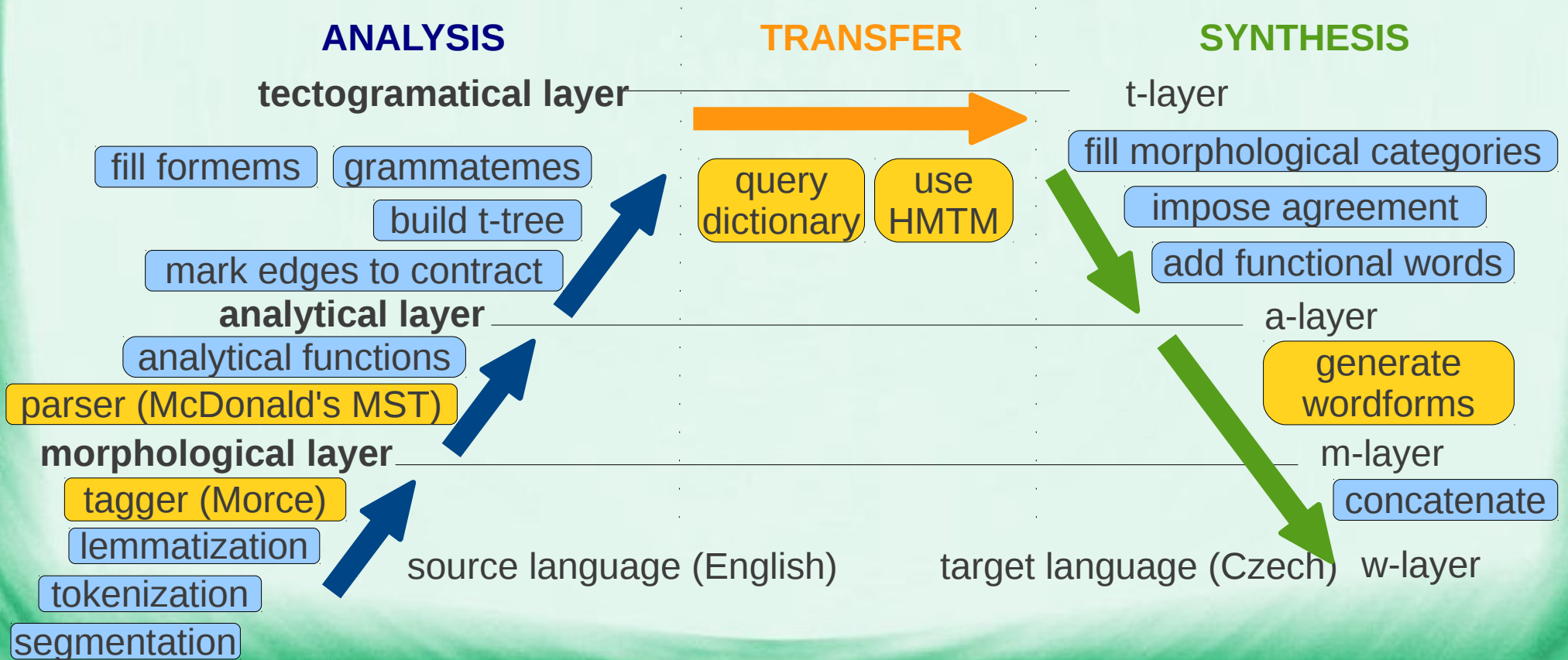
-  easier to use
-  more flexible
-  more langs

\*) Most of the listed languages are only drafts of analysis made by students, not converted to Treex yet. The entire risk as to the quality and performance of the program is with you.

# TectoMT

linguistically motivated MT system (English to Czech pilot)

- deep syntactic (tectogrammatical) transfer
- translation process divided to more than 90 “blocks“
- combining **statistical** and **rule based** blocks

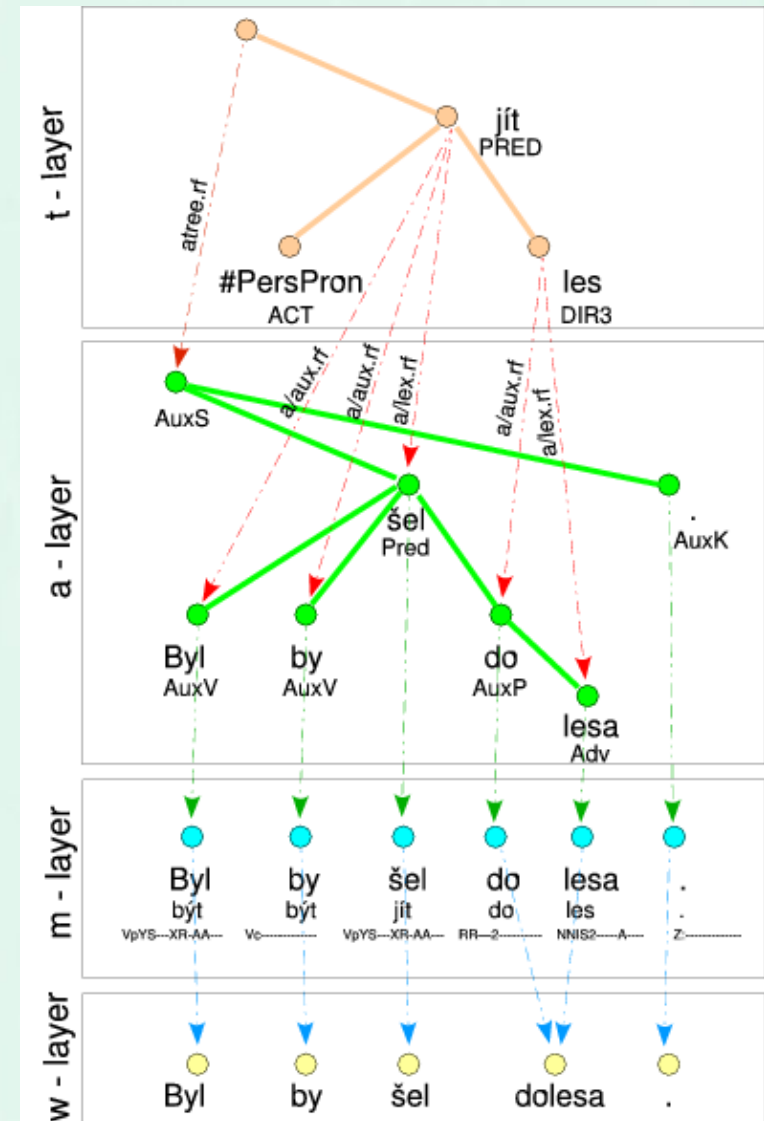


# 4 layers of language description

implemented in Prague Dependency Treebank (PDT)



- **tectogrammatical layer**  
deep-syntactic dependency trees
- **analytical layer**  
surface-syntactic dependency trees, labeled edges
- **morphological layer**  
lemma & POS tag for each word
- **word layer**  
raw (tokenized) text

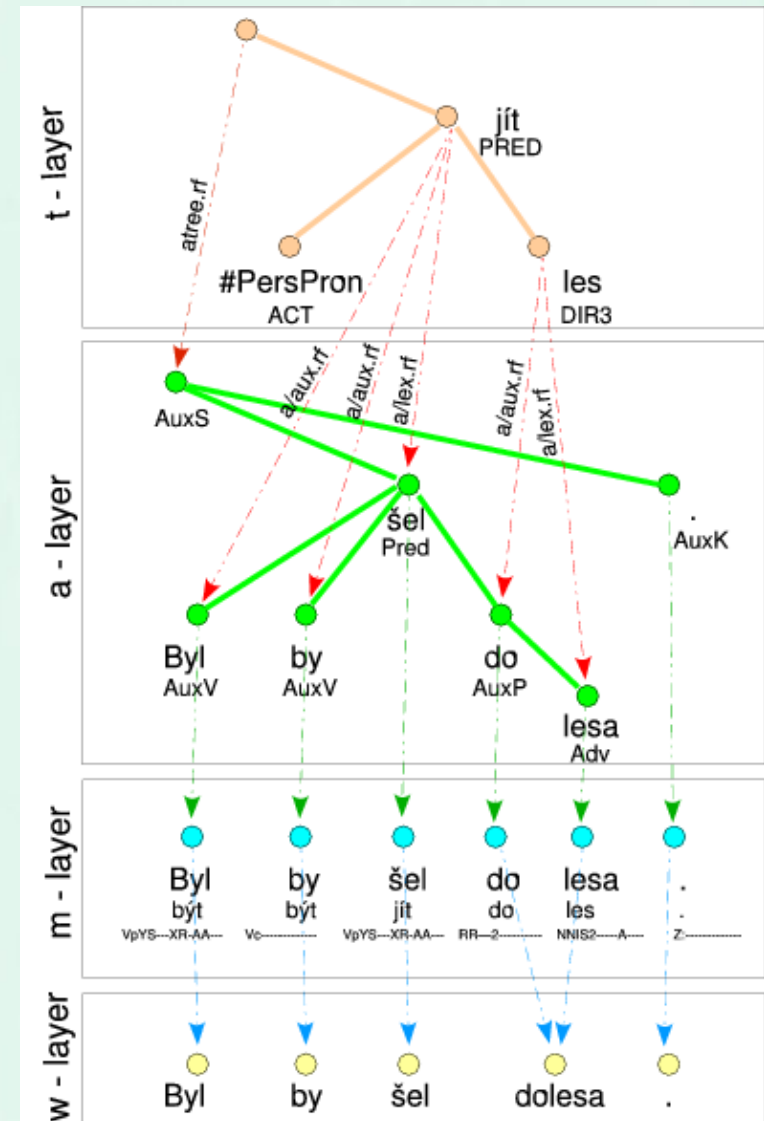


# 4 layers of language description

implemented in Prague Dependency Treebank (PDT)



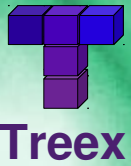
- **tectogrammatical layer**  
deep-syntactic dependency trees
- abstraction from many language-specific phenomena
- autosemantic (meaningful) words  
~ **nodes**
- functional words (prepositions, auxiliaries)  
~ **attributes**
- syntactic-semantic relations (dependencies)  
~ **edges**
- added nodes (e.g. because of pro-drop)
- ...





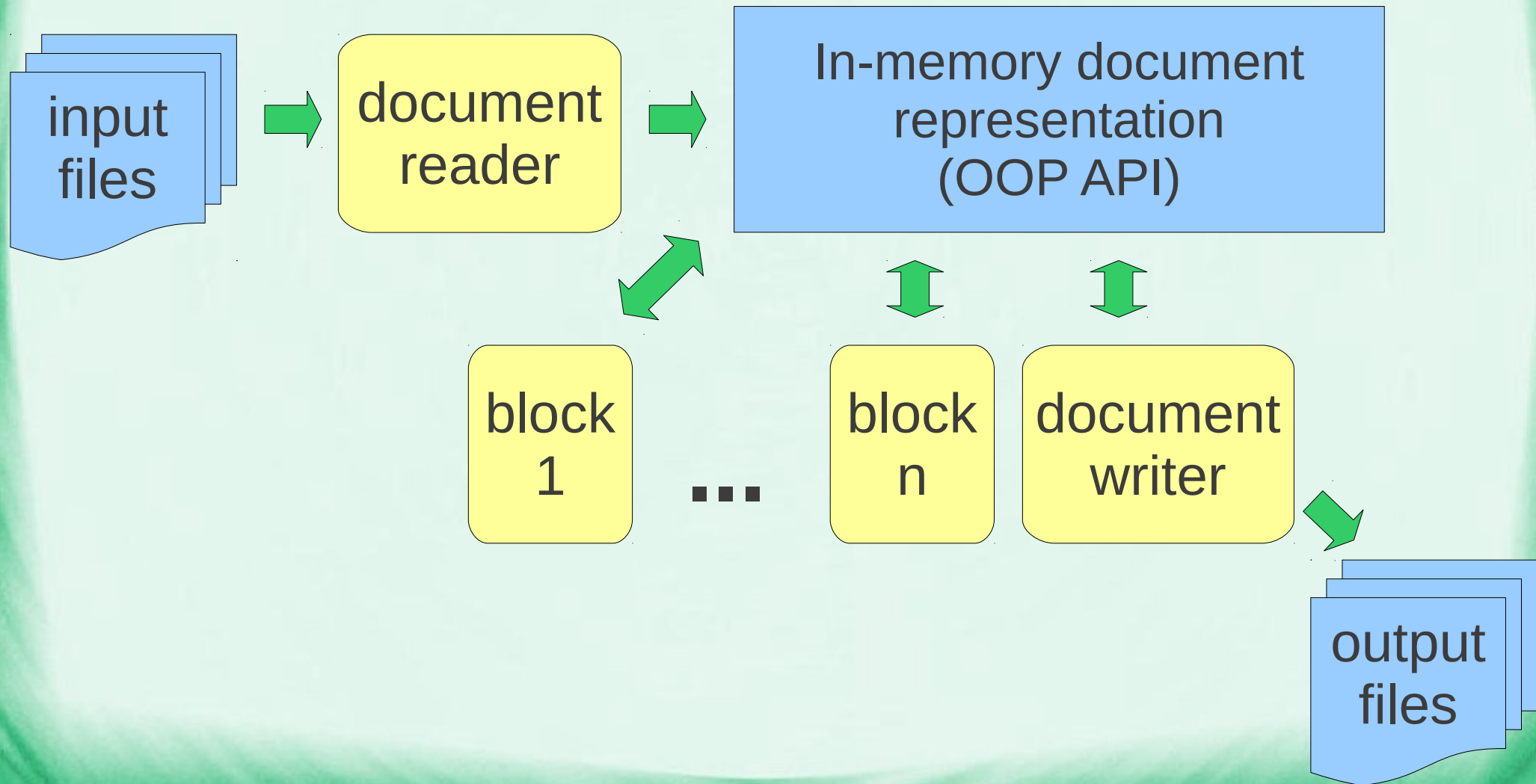
# layers of language description

## implemented in Treex



- Mostly backward compatible adaptations (adding attributes)
  - **formeme** (n:2, n:k+3, v:že+vfin, v:rc, adj:attr)
  - attributes for clauses, is\_passive (→ diathesis),...
- is\_member (for conjuncts on a-layer) is stored with prepositions
  
- All layers stored in **one file**
- A-layer and m-layer merged into one
- Two more layers:
  - P-layer phrase-structure trees
  - N-layer named entities

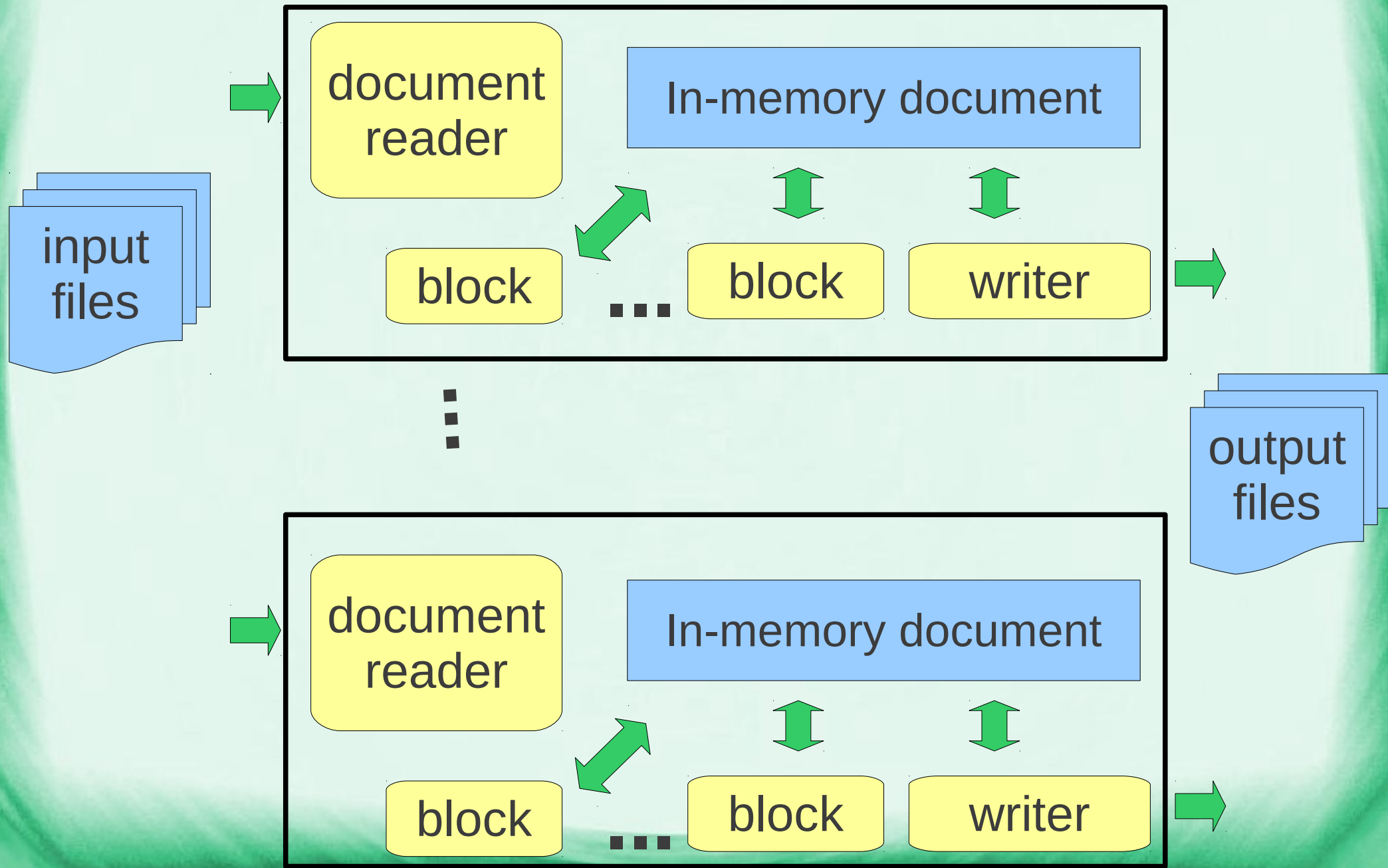
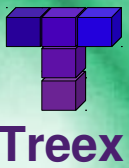
# Treex architecture





# Treex architecture

## parallelization (using SGE cluster)



# Treex architecture processing units



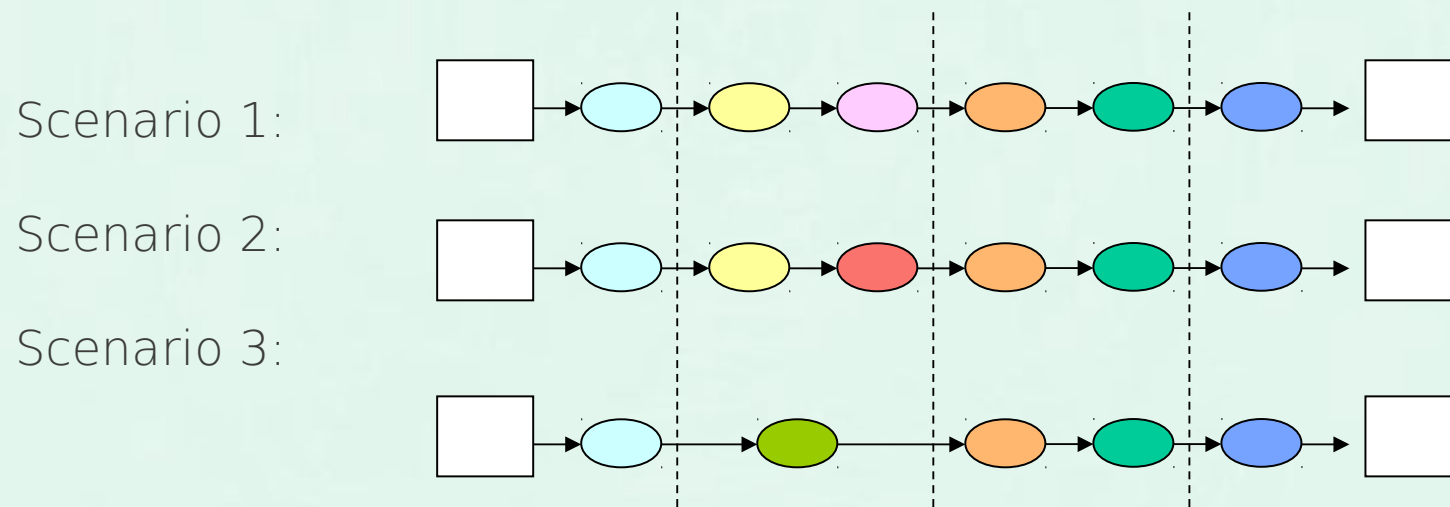
- **block** – elementary processing unit in Treex
  - corresponding to a given NLP subtask
  - one Perl class, saved in one file
- **scenario** – a sequence of blocks
  - saved in plain text files
  - just a list of the blocks' names and their parameters
- **application** – represents an end-to-end NLP task
  - described by a scenario that
    - starts with a **reader** (input conversion)
    - ends with a **writer** (output conversion)
  - Readers can split the input file into more in-memory docs.
  - There are readers&writers for a number of popular formats: plain text, CoNLL, PDT PML, Penn MRG, Tiger...

\* **.treex.gz**

# Treex architecture processing units



Blocks can be easily substituted with an alternative solution.



# Treex architecture processing units



Blocks can be easily substituted with an alternative solution.

Scenario A

**W2A::EN::Segment**

**W2A::EN::Tokenize**

**W2A::EN::TagMorce**

**W2A::EN::Lemmatize**

**W2A::EN::ParseMST**

Scenario B

**W2A::SegmentOnNewLines**

**W2A::EN::TagLinguaEn**

**W2A::EN::Lemmatize**

**W2A::EN::ParseMa1t**

# Treex architecture

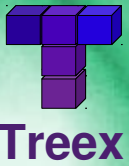
## data units



- **Document**
  - stored in one file
  - sequence of sentences
- **Bundle** (“bundle of trees”)
  - corresponds to one sentence
- **Zone**
  - one for each language (Arabic, Czech, English,...)
  - and optionally a variant (“selectors” src, trg, ref,...)
- **Tree**
  - layer of language description: A, T (plus P, N)
  - m-layer is stored with the a-layer in one tree



# Treex architecture data units



## DOCUMENT

sentence 1

### BUNDLE

Zone en\_src

W-layer

*Peter does not love Mary.*

M-layer

● ● ● ● ●  
Peter do not love Mary  
NNP VBZ RB VBD NNP

A-layer

● ● ● ● ●  
Peter do not love Mary  
Sb AuxV Neg Pred Obj

T-layer

● ● ●  
Peter love Mary  
ACT PRED PAT

Zone cs\_src

W-layer

*Petr nemiluje Marii.*

M-layer

● ● ●  
Petr milovat Marie  
NNMS1 VB-S—3P-NA NNFS4

A-layer

● ● ●  
Petr milovat Marie  
Sb Pred Obj

T-layer

● ● ●  
Petr milovat Marie  
ACT PRED PAT

sentence 2

### BUNDLE

...

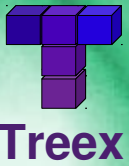
sentence N

### BUNDLE

...



# Treex architecture data units



## DOCUMENT

sentence 1

sentence 2

...

sentence N

### BUNDLE

### BUNDLE

### BUNDLE

Zone en\_src

Zone cs\_src

W-layer

*Peter does not love Mary.*

W-layer

*Petr nemiluje Marii.*

M-layer

● ● ● ●  
Peter do not love Mary  
NNP VBZ RB MD NNP

M-layer

● ●  
Petr miluje Marii  
NNMS1 VB-S NA NNFS4

A-layer

● ● ● ●  
Peter do not love Mary  
Sb AuxV N Pred Obj

A-layer

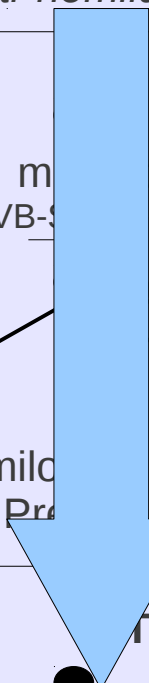
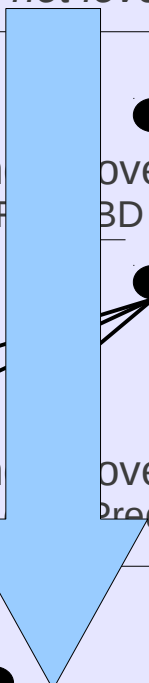
● ● ● ●  
Petr miluje Marii  
Sb Pred Obj

T-layer

●  
Peter love Mary  
ACT PRED PAT

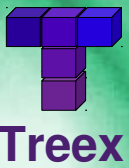
T-layer

● ● ●  
Petr milovat Marie  
ACT PRED PAT



...

# Treex architecture data units



## DOCUMENT

sentence 1

sentence 2

...

sentence N

### BUNDLE

### BUNDLE

### BUNDLE

Zone en\_src

Zone cs\_trg

W-layer

W-layer

*Peter does not love Mary.*

*Petr nemiluje Marii.*

M-layer

M-layer

● ● ● ●  
Peter do not love Mary  
NNP VBZ RB MD NNP

● ● ● ●  
Petr miluje Marii  
NNMS1 VB-IP-NA NNFS4

A-layer

A-layer

● ● ● ● ● ●  
Peter do not love Mary  
Sb AuxV NP-Mod Pred Obj

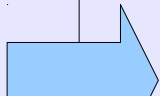
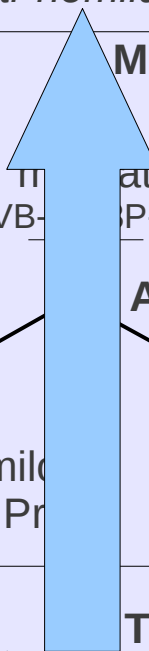
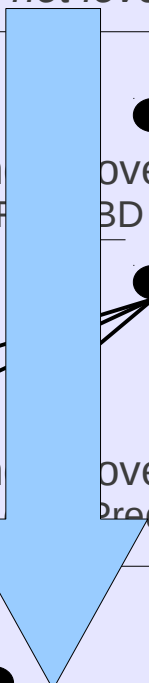
● ● ● ● ● ●  
Petr miluje Marii  
Sb Pr Pred Obj

T-layer

T-layer

● ● ●  
Peter love Mary  
ACT PRED PAT

● ● ●  
Petr milovat Marie  
ACT PRED PAT

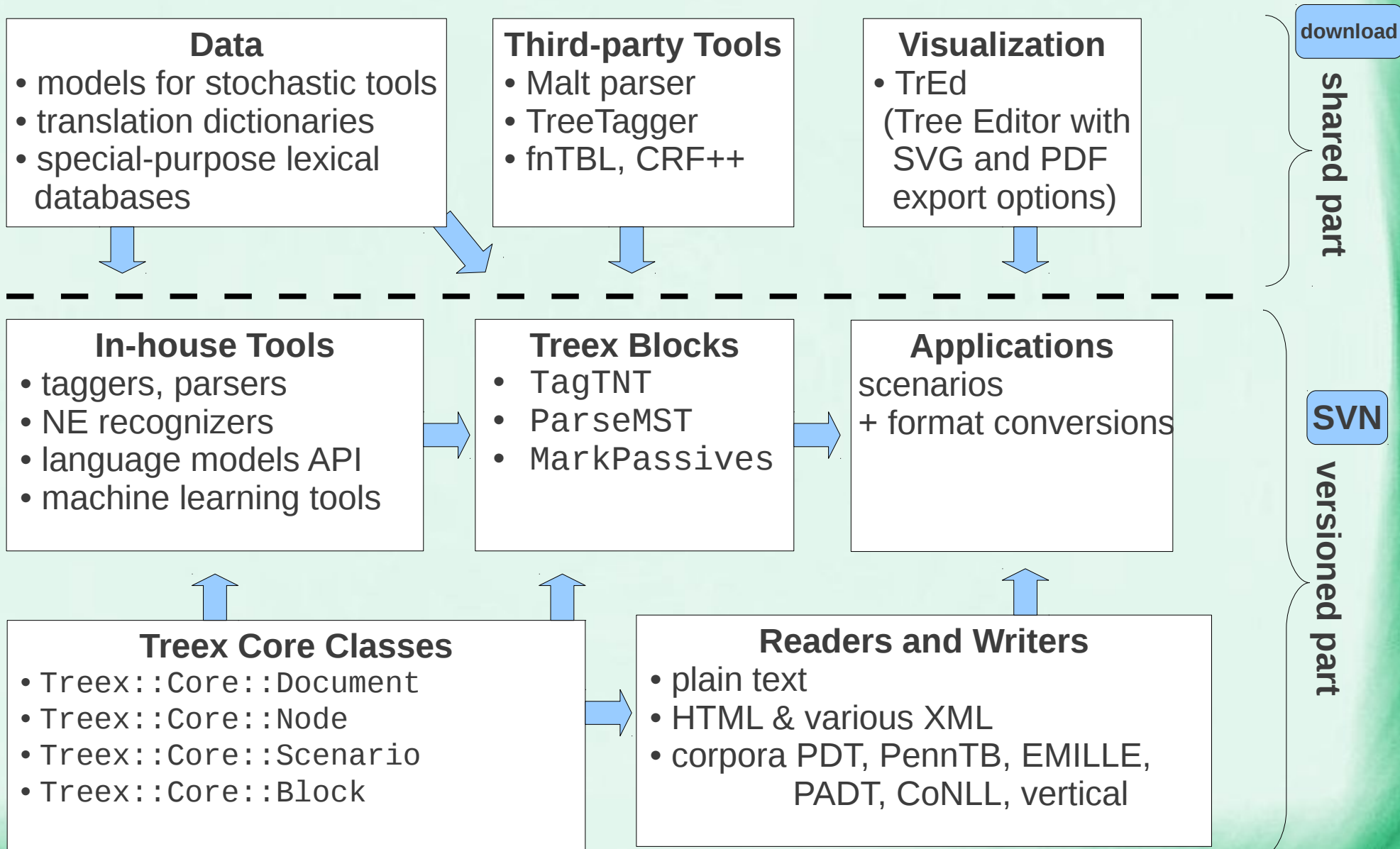


...

# Internals – Design decisions

- Perl (wrappers for binaries, Java,...)
- Linux (some applications platform-independent)
- OOP (Moose)
- Open source (GNU GPL for the versioned part)
- Neutral w.r.t. methodology (statistical, rule-based)
- Multilingual
- Open standards (Unicode, XML)

# Internals – Components



# Internals – Statistics

- Developed since 2005, over ten developers
- Over 400 blocks (140 English, 120 Czech, 60 English-to-Czech, 30 other languages, 50 language independent)
- Taggers (5 English, 3 Czech, 1 German and Russian, Tamil)  
Parsers (Dep. 2 English, 3 Czech, 2 German; Const. 2 English)  
Named Entity Recognizers (2 Czech, 1 English)
- Speed example: Best version of English-to-Czech MT  
1.2 seconds per sentence plus 90 seconds loading,  
with 20 computers in cluster: 2000 sentences in 4 min



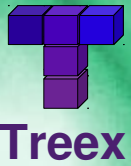
# Future plans

- CPAN release
    - Treex::Core done
    - Treex::EN soon
    - Treex::CS, Treex::DE,...
  - Lot of “invisible” work (testing)
  - Manual, tutorial, FAQ, demos
  - Adding new document readers more easily
  - Integrating more tools, more languages
  - Web services? Alternative parallelization?
  - Your applications, your requests...
- Tomáš Kraut



# Conclusion

## Treex main properties



- emphasized efficient development, modular design and reusability
- stratificational approach to the language
- unified object-oriented interface for accessing data structures
- comfortable development

# TrEd visualization

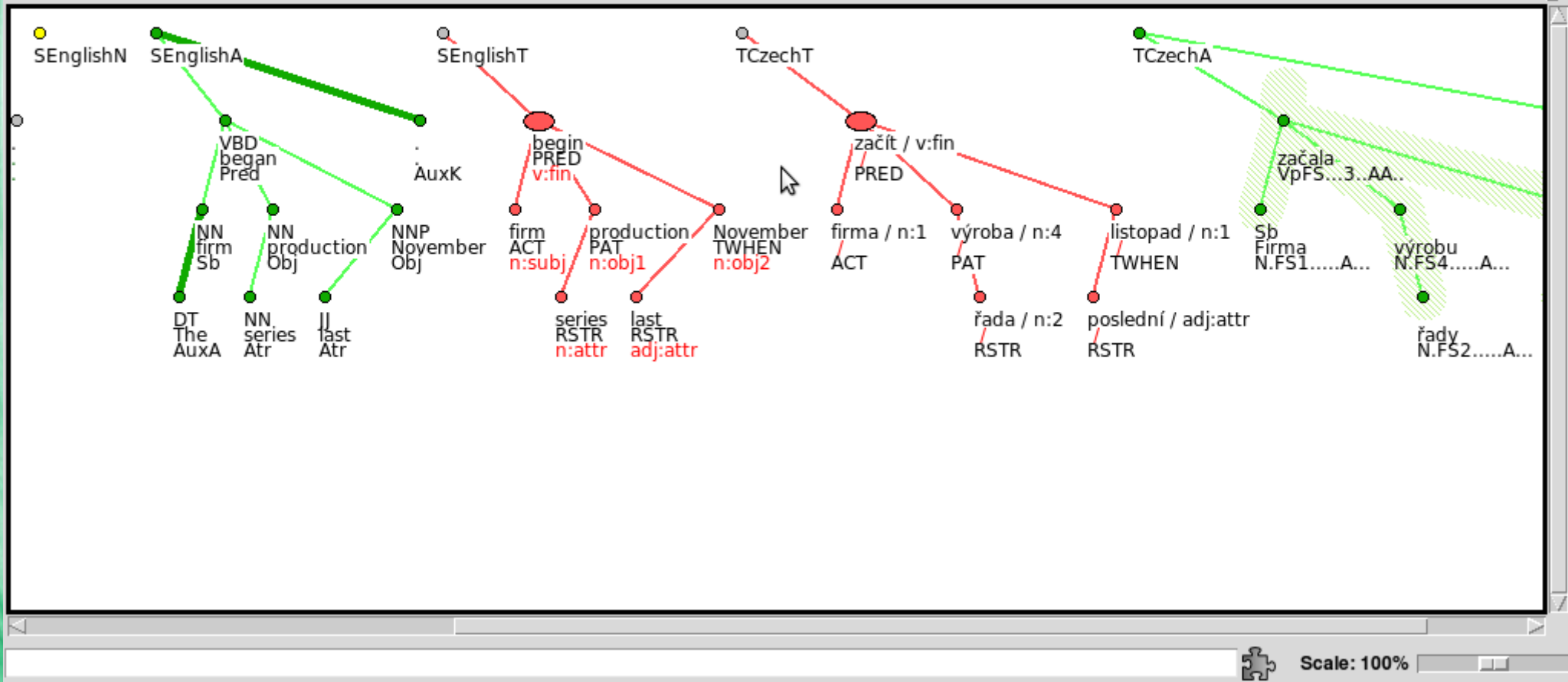
## translation

File Node Tree View Macros Setup Help Mode: TectoMT\_TredMacros

Style: TectoMT

The firm began series production last November.  
 Sériovou výrobu firma rozjela loni v listopadu.  
 Firma začala výrobu řady poslední listopad.

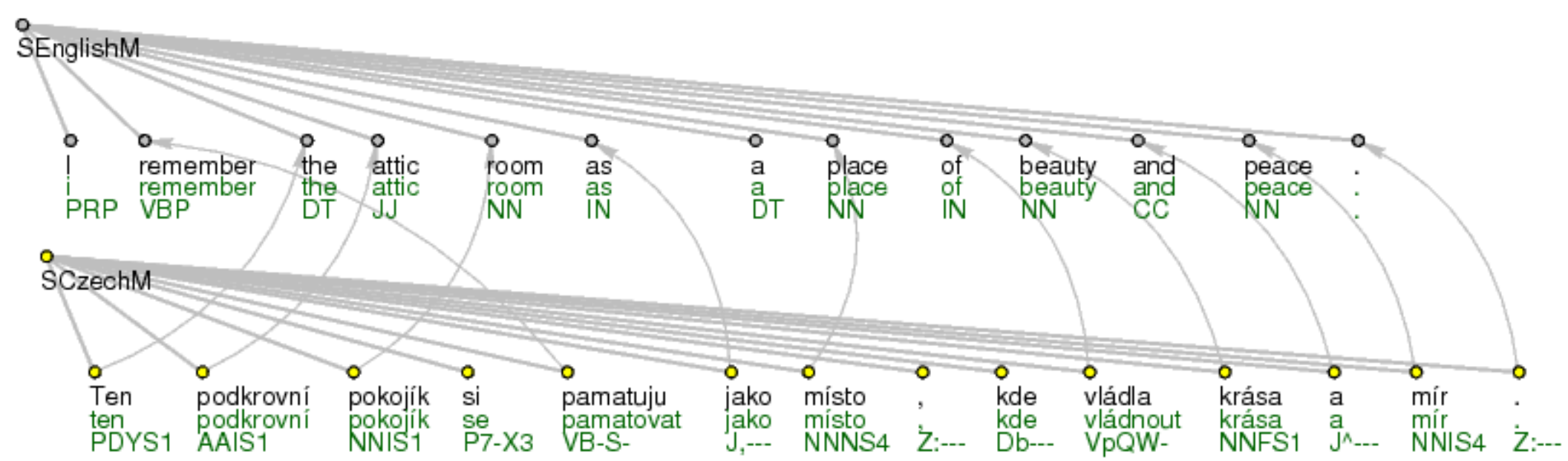
2/50



Scale: 100%

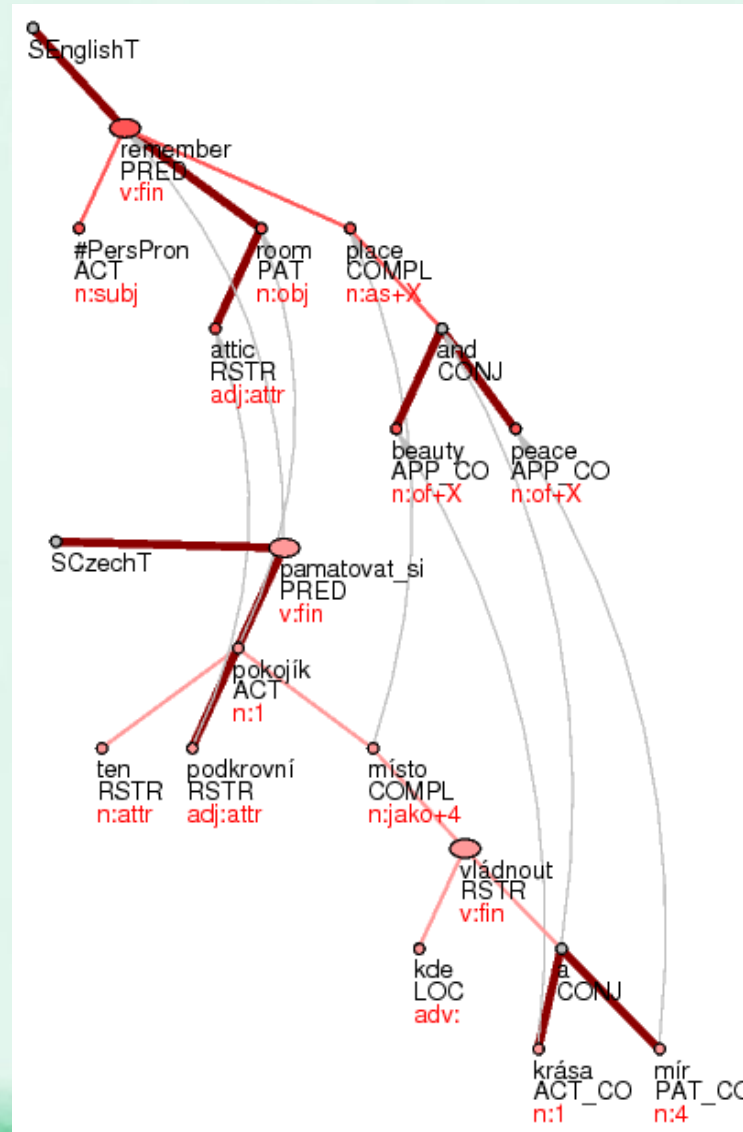
# TrEd visualization

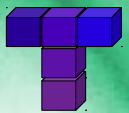
word alignment on the morphological layer



# TrEd visualization

word alignment on the tectogrammatical layer





# TrEd visualization

## named entities

File Node Tree View Macros Setup Help

Mgde: TectoMT\_TredMacros

Style: TectoMT

4/14

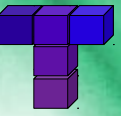
Tři utonulí jsou z Jeseníku nad Odrou na Novojičínsku a jedna žena utonula v Novém Jičíně-Žilině.

The visualization shows a sentence with its grammatical structure and named entities. The sentence is: "Tři utonulí jsou z Jeseníku nad Odrou na Novojičínsku a jedna žena utonula v Novém Jičíně-Žilině." The structure is a tree where nodes represent grammatical functions and named entities. Named entities are highlighted with red hatched boxes. The entities shown are: "Novojičínsko" (type: gro), "Nový Jičín", "Žilina", "Jeseník nad Odrou", "Novojičínsko", "Nový Jičín", "Žilina", "Novojičínsko", "Nový Jičín", "Žilina". The structure is a tree where nodes represent grammatical functions and named entities. The entities are highlighted with red hatched boxes. The structure is a tree where nodes represent grammatical functions and named entities. The entities are highlighted with red hatched boxes.

Named entity: normalized name=Novojičínsko type=gro (oblast - okolí města)

Scale: 100%





# Block example – SVO to SOV code

```
package Tutorial::Svo2SovSolution;  
use Moose;  
use Treex::Core::Common;  
extends 'Treex::Core::Block';
```

```
sub process_anode {  
    my ( $self, $a_node ) = @_;  
    if ( $a_node->tag =~ /^V/ ) {          # verb found  
        foreach my $child ( $a_node->get_echildren() ) {  
            if ( $child->afun eq 'Obj' ) {  # object found  
                # Move the object and its subtree so it precedes the verb  
                $child->shift_before_node($a_node);  
            }  
        }  
    }  
    return;  
}
```

Treex core

Treex convention

Perl keyword/convention

# Thank you

Cooperation is welcomed.



<http://ufal.mff.cuni.cz/tectomt>



# Thank you

TreeX is growing!



<http://ufal.mff.cuni.cz/tectomt>