An Open Source Infrastructure for Language Technology. Design parameters for such an infrastructure, and Giellatekno as an example

Sjur Moshagen
Divvun, UiT

Trond Trosterud
Giellatekno, UiT

Contents

Introduction

- Reykjavik 2004: Vigdis Finnbogadóttir's birthday
  - There, I compared language technology and the open source programming movement

Introduction

- Reykjavik 2004: Vigdis Finnbogadóttir's birthday
  - There, I compared language technology and the open source programming movement
  - My bottom line was: Lanuage technology is different.

Introduction

- Reykjavik 2004: Vigdis Finnbogadóttir's birthday
  - There, I compared language technology and the open source programming movement
  - My bottom line was: Lanuage technology is different.
  - It turned out that in a very important sense I was wrong

Introduction

- Reykjavik 2004: Vigdis Finnbogadóttir's birthday
  - There, I compared language technology and the open source programming movement
  - My bottom line was: Lanuage technology is different.
  - It turned out that in a very important sense I was wrong
- This talk discusses how wrong I was, and why.

Examples of open infrastructure for language technology

Giellatekno infra


Tromsø
S
F

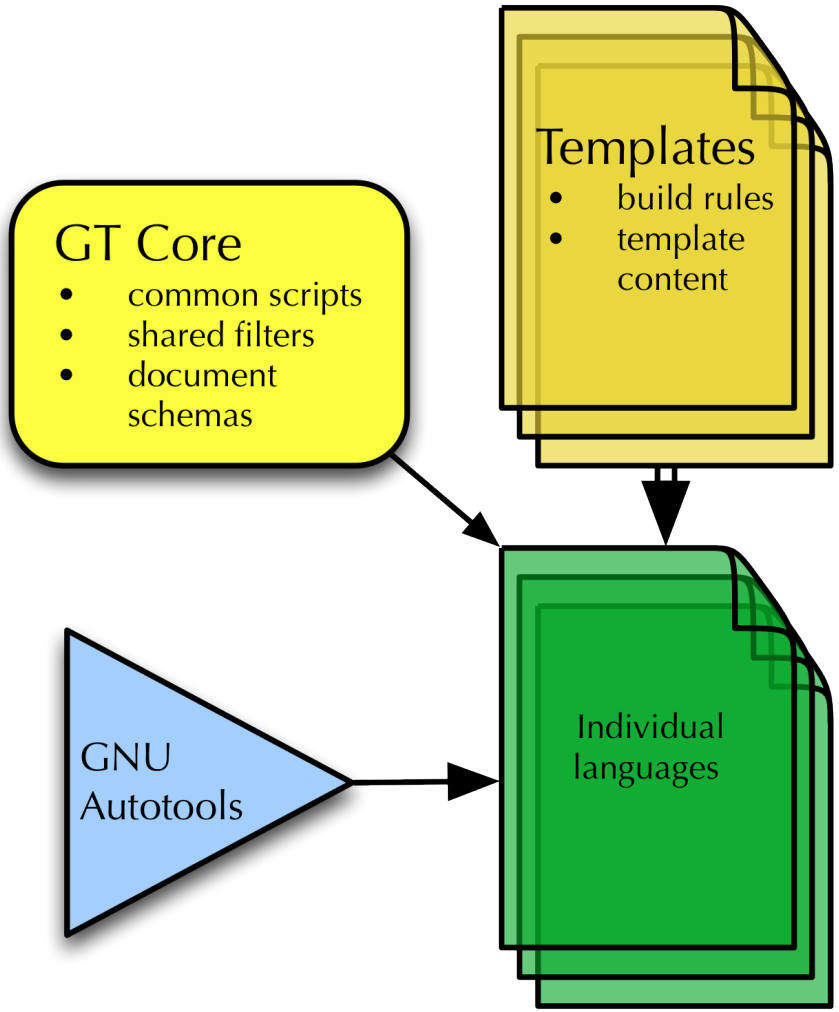
Giellatekno?

`http://giellatekno.uit.no`

The core idea of the Giellatekno infra

1. Uniform infra for all lgs
2. Cut the S curve
3. Plug in your language and get out whatnot
4. The message being – *Join in!*

So, what do we do, and how?



GT Core

- core and shared functionality, resources and source code
- templates for build instructions and template source files for the languages
- templates for shared linguistic content / source code

```
!Divvun & Giellatekno - open source grammars for Sámi and other languages
!Copyright © 2000-2010 The University of Tromsø & the Norwegian Sámi Parliament
!http://giellatekno.uit.no & http://divvun.no
!
!This program is free software ; you can redistribute and/or modify this file under the terms
!of the GNU General Public License as published by the Free Software Foundation, either
!version 3 of the License, or (at your option) any later version. The GNU General Public
!License is found at http://www.gnu.org/licenses/gpl.html. It is also available in the file
!$GTHOME/LICENSE.txt.
!
!Other licensing options are available upon request, please contact giellatekno@hum.uit.no or
!divvun@samediggi.no This file contains Cyrillic proper nouns that should be merged with all
!URJ and other languages using the Cyrillic alphabet. INITIALLY the file will contain surnames,
!given names and patronymics  that cannot be regularly derived from male given names. A dis-
!tinction should be made for use of _ë_ and _e_ spell-relax [ë (->) e] will be used where necessary

LEXICON urj-Cyrl-ProperNouns
Аалунд:Аалунд CYRL-CONS_SUR "Z" ;
Абабков:Абабков Deriv-RUS-B_SURMAL "Z" ;
Абаджев:Абаджев Deriv-RUS-B_SURMAL "Z" ;
Абаджян:Абаджян CYRL-CONS_SUR "Z" ;
Абаев:Абаев Deriv-RUS-B_SURMAL "Z" ;
Абазадзе:Абазадзе CYRL-VOW_SUR "Z" ;
Абазаев:Абазаев Deriv-RUS-B_SURMAL "Z" ;
Абазашвили:Абазашвили CYRL-VOW_SUR "Z" ;
Абазов:Абазов Deriv-RUS-B_SURMAL "Z" ;
Абазян:Абазян CYRL-CONS_SUR "Z" ;
Абаимов:Абаимов Deriv-RUS-B_SURMAL "Z" ;
```

GT Core 2

- Language independent template in gtcore
  - content merged with all languages when template is changed
  - thus languages always have updated build instructions and template source files

GT Core 2

- Language independent template in gtcore
  - content merged with all languages when template is changed
  - thus languages always have updated build instructions and template source files
- All languages get all new functionality and features along the road
  - develop once, deploy everywhere
  - (the linguistic content of course needs to be built in each case)

GT Core 2

- Language independent template in gtcore
  - content merged with all languages when template is changed
  - thus languages always have updated build instructions and template source files
- All languages get all new functionality and features along the road
  - develop once, deploy everywhere
  - (the linguistic content of course needs to be built in each case)
- GT Core is configured and installed in the regular way using autotools.

Dependencies

- hfst
  - Helsinki Finite State Transducers - morphology
- vislcg3
  - CG = Constraint Grammar - rule-based syntax tagging
- Autotools
- standard packages and tools

hfst

`http://hfst.sourceforge.net`

vislcg3

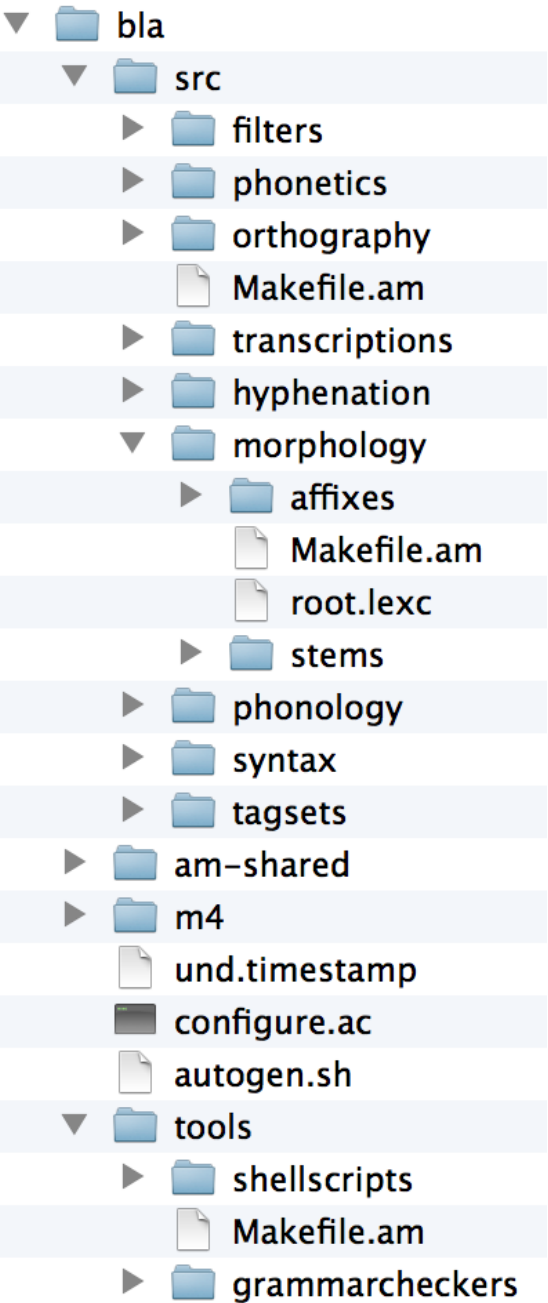
`http://beta.visl.sdu.dk/cg3.html`

Getting started

http://giellatekno.uit.no/doc/infra/GettingStarted.html

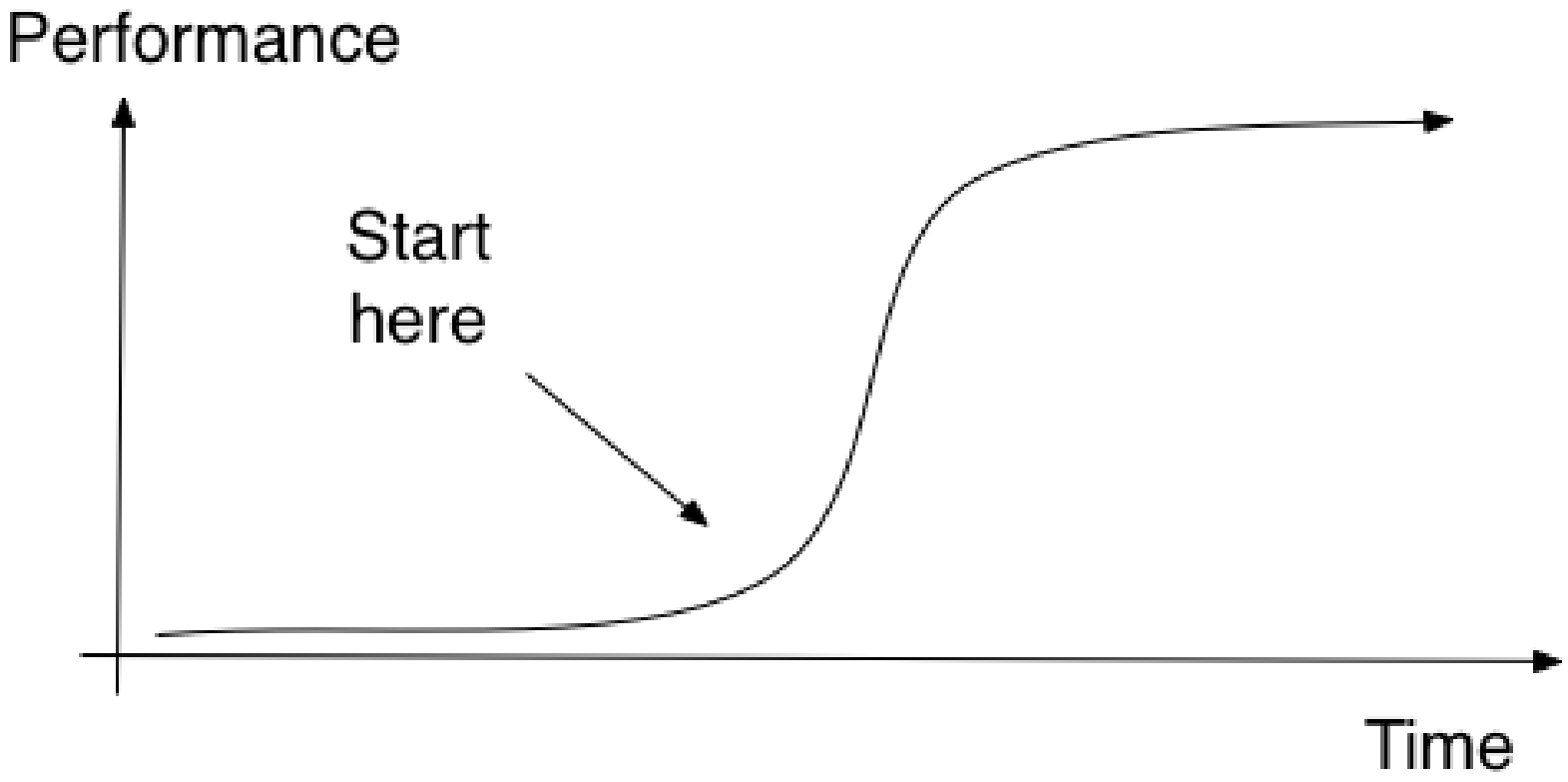
Download the repository

`http://giellatekno.uit.no/doc/tools/docu-svn-user.html`

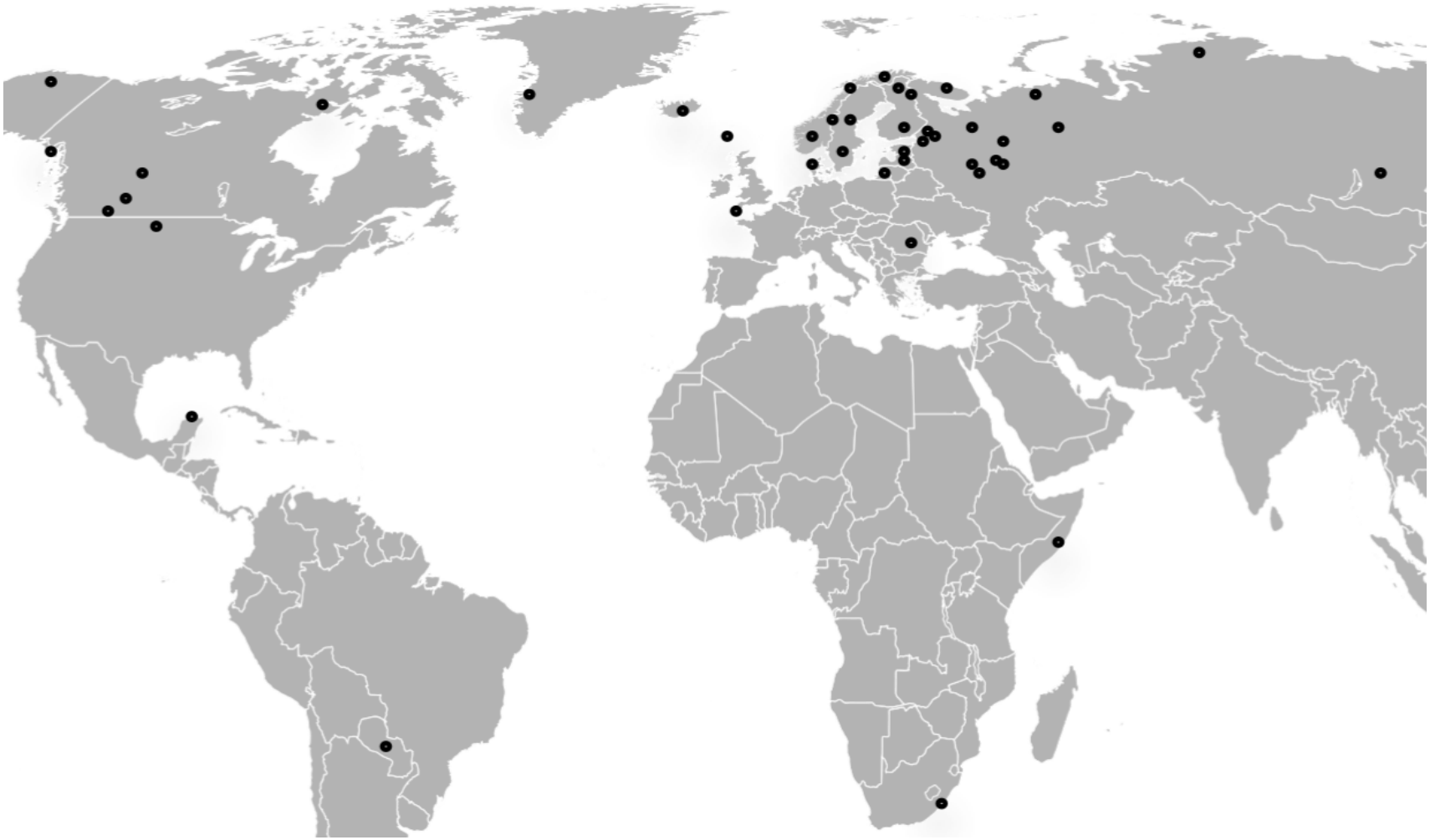

bla
src
filters
phonetics
orthography
Makefile.am
transcriptions
hyphenation
morphology
affixes
Makefile.am
root.lexc
stems
phonology
syntax
tagsets
am-shared
m4
und.timestamp
configure.ac
autogen.sh
tools
shellscripts
Makefile.am
grammarcheckers

«So, what is in it for me?»

«So, what is in it for me?»



Giellatekno languages - 44 as of now



Giellatekno languages

- each language build instructions and automated tests

Giellatekno languages

- each language build instructions and automated tests
- each language can build the same set of linguistic tools

Giellatekno languages

- each language build instructions and automated tests
- each language can build the same set of linguistic tools
  - ... but not all languages have source files developed for all tools

Giellatekno languages

- each language build instructions and automated tests
- each language can build the same set of linguistic tools
  - ... but not all languages have source files developed for all tools
- tools and tool sets can be turned on or off via configuration options

What we presently build:

- basic package (on by default except hfst):
  - `build with Xerox: yes`
  - `build with HFST: yes`
  - `analysers enabled: yes`
  - `generators enabled: yes`
  - `syntactic processing enabled: yes`
  - `yaml tests enabled: yes`
  - `generated documentation enabled: yes`

What we presently build:

- proofing tools (off by default):
  - `spellers enabled: no`
  - `hfst speller fst's enabled: no`
  - `voikko speller enabled: no`
  - `foma speller enabled: no`
  - `grammar checker enabled: no # (experimental, only one language)`

What we presently build:

- proofing tools (off by default):
  - `spellers enabled: no`
  - `hfst speller fst's enabled: no`
  - `voikko speller enabled: no`
  - `foma speller enabled: no`
  - `grammar checker enabled: no # (experimental, only one language)`
- specialised fst's (off by default):
  - `phonetic/IPA conversion enabled: no`
  - `dictionary fst's enabled: no`
  - `Oahpa transducers enabled: no`
  - `Apertium transducers enabled: no`

Apertium machine translation

http://wiki.apertium.org

Apertium

- http://wiki.apertium.org/wiki/User:Francis_Tyers/An_MT_system_in_one_thousand_steps

Possible projects for the Open Source Community

User and programming interfaces

Interface in the age of multilingualism

- bad language UI in LibreOffice/OpenOffice

Interface in the age of multilingualism

- bad language UI in LibreOffice/OpenOffice
  - => only show enabled languages and document languages
- bad language selection/integration architecture in LO/OOo
  - => language features should be independent modules installed via plugins, not at compile time (more like Windows 8)

Interface in the age of multilingualism

- bad language UI in LibreOffice/OpenOffice
  - => only show enabled languages and document languages
- bad language selection/integration architecture in LO/OOo
  - => language features should be independent modules installed via plugins, not at compile time (more like Windows 8)
  - must include language/locale code and at least one language name

Interface in the age of multilingualism

- bad language UI in LibreOffice/OpenOffice
  - => only show enabled languages and document languages
- bad language selection/integration architecture in LO/OOo
  - => language features should be independent modules installed via plugins, not at compile time (more like Windows 8)
  - must include language/locale code and at least one language name
  - can include e.g. keyboards, fonts, proofing tools, index and search components, etc.

Langtech to the people: Packages

- Packages for the language developers / linguists
- Packages for the the distributors and end users
- Targeted package systems

Packages for the language developers / linguists

- Goal: prepare the system for what is needed to do linguistic development
- install and set up the gtcore and all its dependencies
- Example (using MacPorts as example package manager):
  - `sudo port install gtcore`
  - svn checkout of the language(s) one wants to work with

Packages for the the distributors and end users

- Goal: make the language tools available to end users
- Example - Russian:
- install and set up a package based upon the Russian resources, and all dependencies, including `gtcore`
  - `sudo port install gtlang-rus`

Targeted package systems

- Debian
- RPM?
- MacPorts
- a Windows package manager
  - (or would other channels be better to reach Windows users?)

Integration tasks

- speller integrated in all text interfaces
  - (spell checker everywhere)
- Voikko+Hfst-ospell integrated parallel to other spellers
  - (ispell etc. is not enough)

New features

- indexing and searching (lemmatising/stemming) for all languages

New features

- indexing and searching (lemmatising/stemming) for all languages
- grammar checker technology for real grammar checking

New features

- indexing and searching (lemmatising/stemming) for all languages
- grammar checker technology for real grammar checking
- other writing tools:
  - automatic hyphenation
  - predictive writing
  - grammatical editing (swap NP order, change person or number of pronoun and get the verb(s) automatically updated)

New features

- LT support in localisation tools
  - get localised UI faster and for more languages) -
  - MT,
  - linguistically intelligent TM

New features

- LT support in localisation tools
  - get localised UI faster and for more languages) -
  - MT,
  - linguistically intelligent TM
- Linux as a platform for language learning

New features

- LT support in localisation tools
  - get localised UI faster and for more languages) -
  - MT,
  - linguistically intelligent TM
- Linux as a platform for language learning
- speech synthesis for all

Licensing

Licensing

- The vislcg3 license

Problems and solutions

- reimplementation or another license

Present status

- Divvun/GT - ok
- Hfst - ok (without Foma and sfst)
- vislcg3 (GC, synt. analysis) - NOT ok

Conclusions

The buzzwords

- open source
- language independent
- flexible
- easily accessible
- easily extendable
- standards-based

Conclusion

- we have the language technology and linguistic resources

Conclusion

- we have the language technology and linguistic resources
- we would like to cooperate with other parts of the free/libre open source movement to turn these into end user products to the best of all, for as many languages as possible

Conclusion

- we have the language technology and linguistic resources
- we would like to cooperate with other parts of the free/libre open source movement to turn these into end user products to the best of all, for as many languages as possible
- http://giellatekno.uit.no