WP 2: Grammar Developer's Tools

Aarne Ranta

MOLTO Project Review, Luxembourg, 15 March 2011

Overview

Ongoing work

Future work

Dissemination

Overview

Objective

Tool for: building **domain-specific grammar-based multilingual translation systems**.

- unlimited set of simultaneous languages
- demonstrated with 15 languages

User expertise: **non-linguist domain expert**

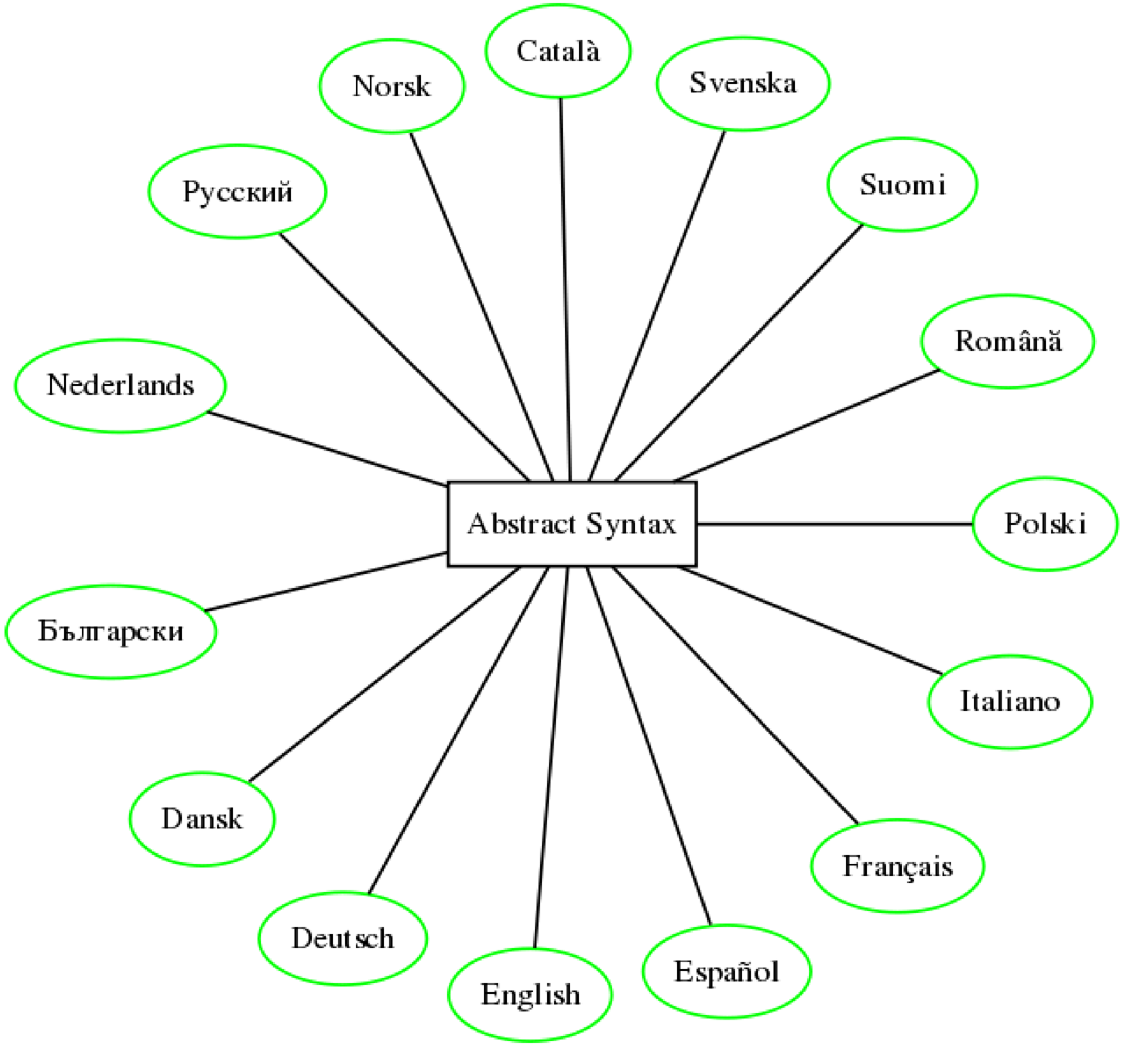
Prior technology

GF = Grammatical Framework, a programming language for multilingual grammars

GF Grammar Compiler, from high-level code to PGF virtual machine (Portable Grammar Format)

GF Shell, an interactive command-line-based environment

GF Resource Grammar Library, library with syntax and morphology for 15 languages



MOLTO languages

Challenge: measurable goals

Scale up production of domain interpreters

- from 100's to 1000's of words
- from GF experts to domain experts and translators
- from months to days
- writing a grammar $\approx$ translating a set of examples

Deliverables

2.1	**GF Grammar Compiler API**	**M12**
2.2	Grammar IDE	M18
2.3	Grammar Tool Manual and Best Practices	M24

Notice: the call doesn't allow the development of linguistic resources. Thus we build on the existing Resource Grammar Library; but the Library is developed in a separate project.

Ongoing work

The Grammar Compiler API

Portable infrastructure for multilingual grammar development:

- project management (up to hundreds of modules)
- rule extraction (from lexical databases, ontologies, text examples)
- compilation (separately per module)
- testing (unit, regression, random, properties)
- maintenance (bug fixing, extensions, refactoring)

The API and the IDE

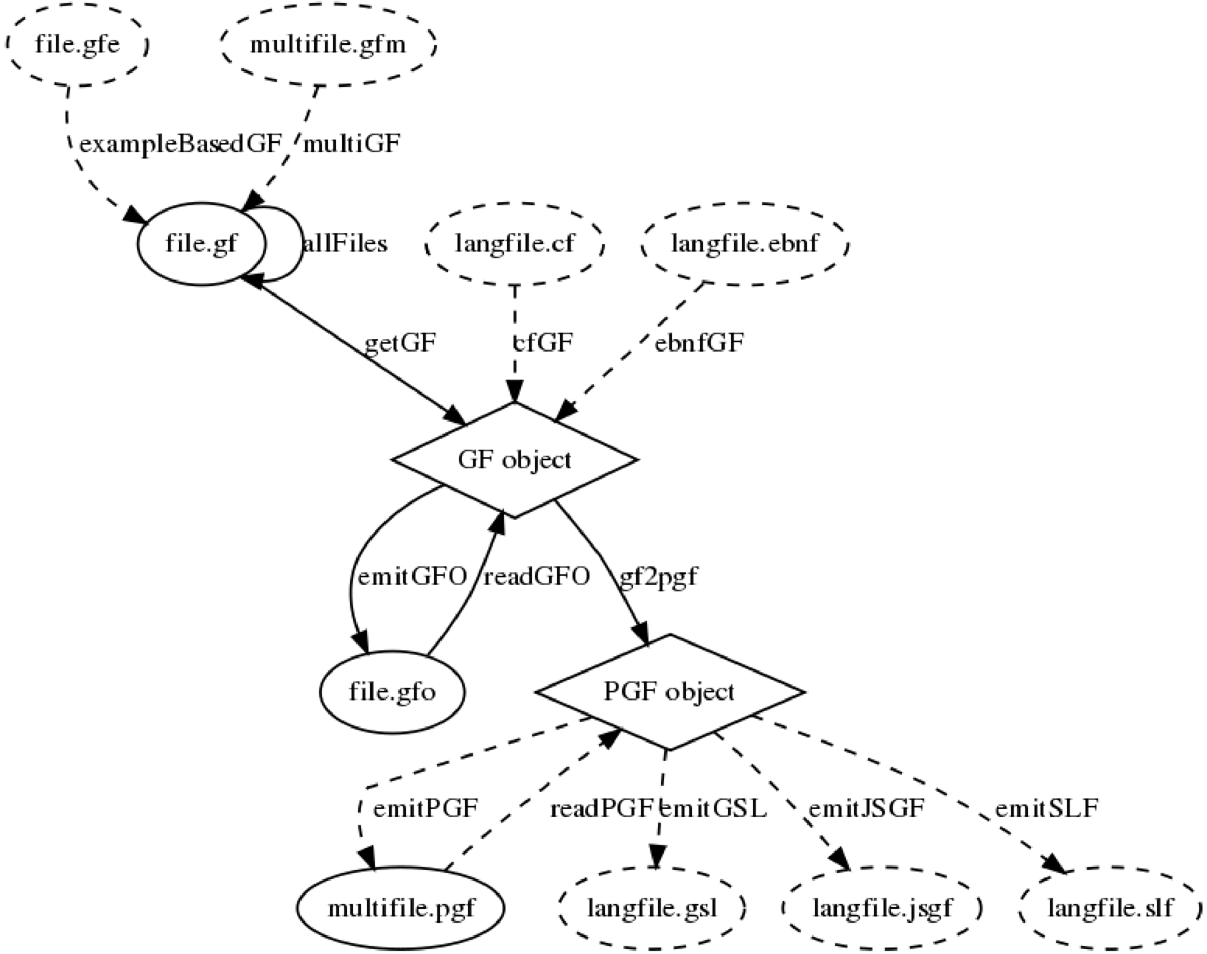
API: a **library** in two formats:

- **Desktop API**: types and functions available for Haskell programs
- **Web API**: functionalities available via server calls

IDE: a **user software system** reproducible in many formats:

- Desktop system (Emacs, Eclipse, Xcode, ...), downloaded and installed locally
- Web-based system, "in the cloud"

Compilation work flow



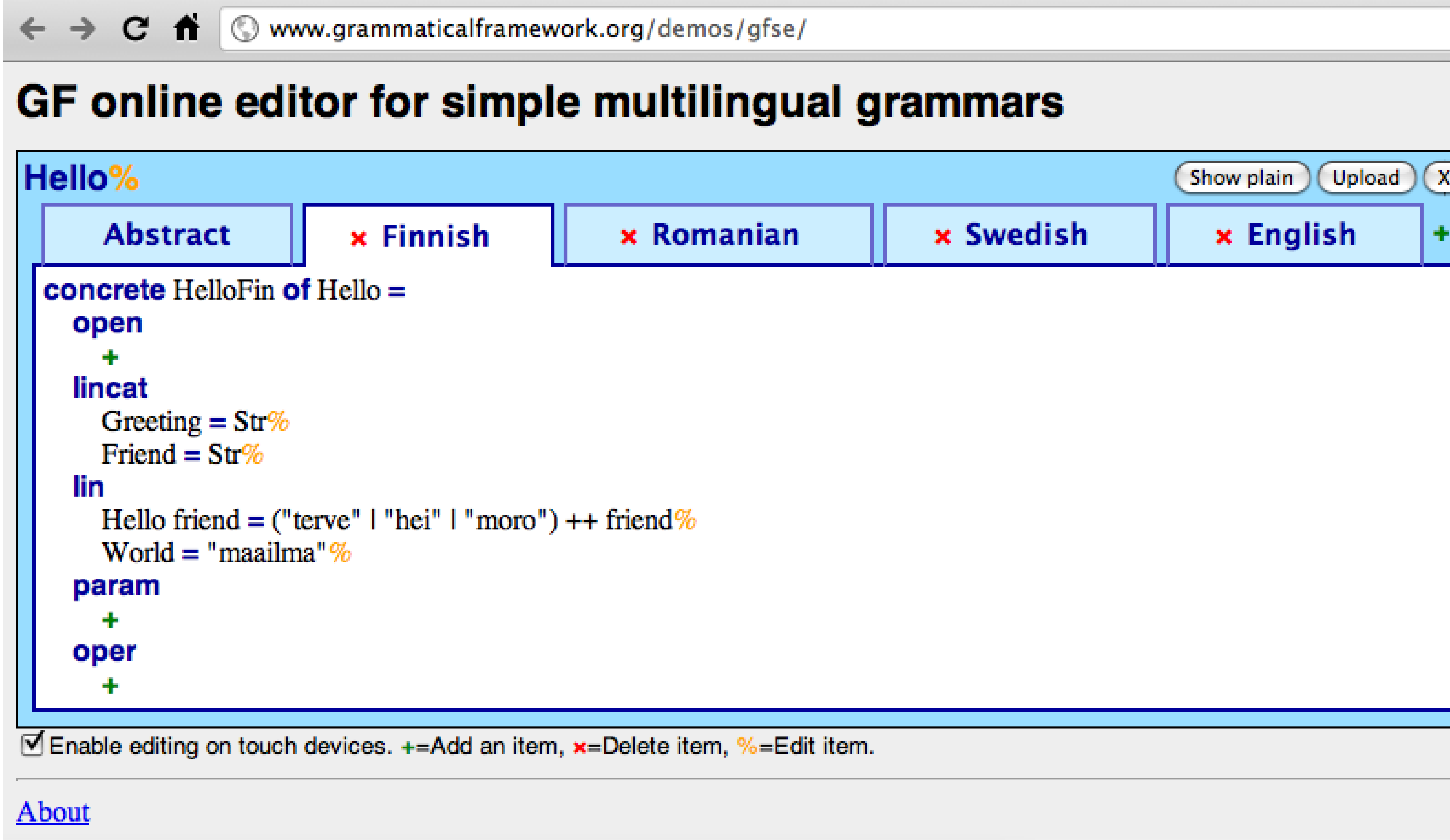
The Haskell API

```
exBasedGF :: FilePath -> IO GF
multiGF   :: FilePath -> IO GF
getGF     :: FilePath -> IO GF
cfGF      :: FilePath -> IO GF
ebnfGF    :: FilePath -> IO GF
emitGFO   :: GF -> IO ()
readGFO   :: FilePath -> IO GF
gf2pgf    :: GF -> PGF
emitPGF   :: PGF -> IO ()
readPGF   :: FilePath -> IO PGF
emitJSGF  :: PGF -> IO ()
emitSLF   :: PGF -> IO ()

checkTerm      :: GF -> Module -> Term -> Type -> Check Term
checkJudgement :: GF -> Module -> Judgement -> Check Judgement
evalTerm       :: GF -> Term -> Check Term
showOpers      :: GF -> Type -> [(Name,Type)] -- opers with desired value type
addJudgement   :: GF -> Module -> Judgement -> Check Module
addModule      :: GF -> Module -> Check GF
addConstruct   :: GF -> Judgement -> [Judgement] -> Check GF
```

The first IDE in the cloud

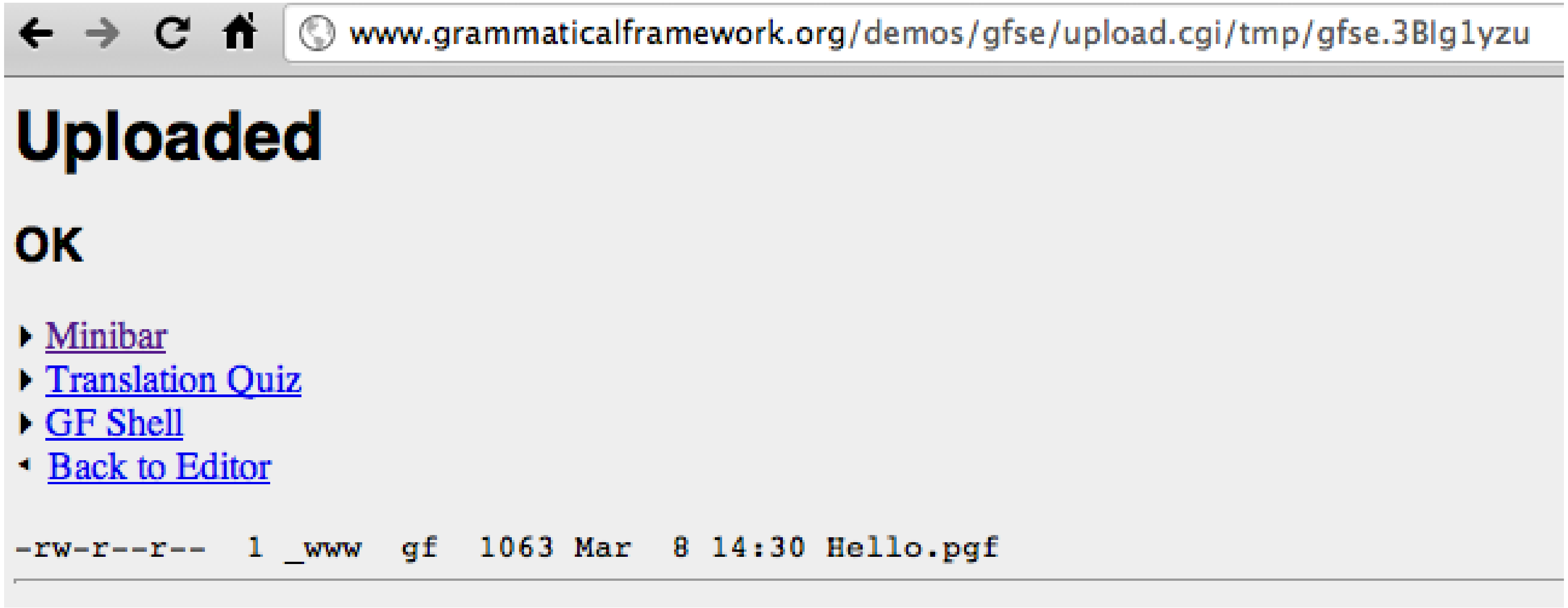
http://www.grammaticalframework.org/demos/gfse/



GF online editor for simple multilingual grammars

```
concrete HelloFin of Hello =
   open
      +
   lincat
      Greeting = Str%
      Friend = Str%
   lin
      Hello friend = ("terve" | "hei" | "moro") ++ friend%
      World = "maailma"%
   param
      +
   oper
      +
```

☑ Enable editing on touch devices. +=Add an item, ×=Delete item, %=Edit item.



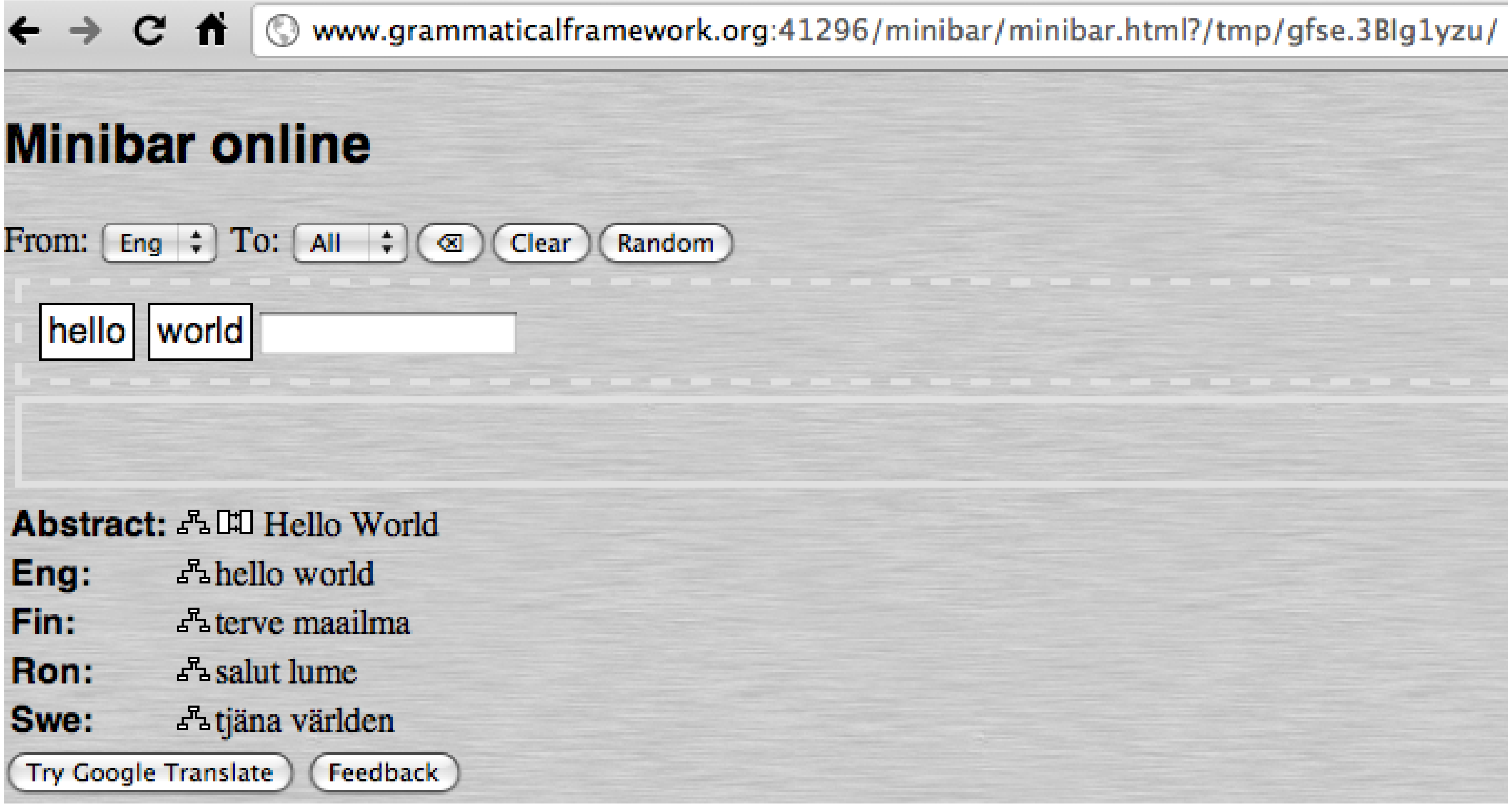
www.grammaticalframework.org/demos/gfse/upload.cgi/tmp/gfse.3Blg1yzu

Uploaded

OK

- Minibar
- Translation Quiz
- GF Shell
- Back to Editor

```
-rw-r--r--  1 _www  gf  1063 Mar  8 14:30 Hello.pgf
```



www.grammaticalframework.org:41296/minibar/minibar.html?/tmp/gfse.3Blg1yzu/
Eng
All
⌫
Clear
Random
Try Google Translate
Feedback

Minibar online

From: Eng To: All

hello world

Abstract: Hello World
Eng: hello world
Fin: terve maailma
Ron: salut lume
Swe: tjäna världen

The GF Resource Grammar Library

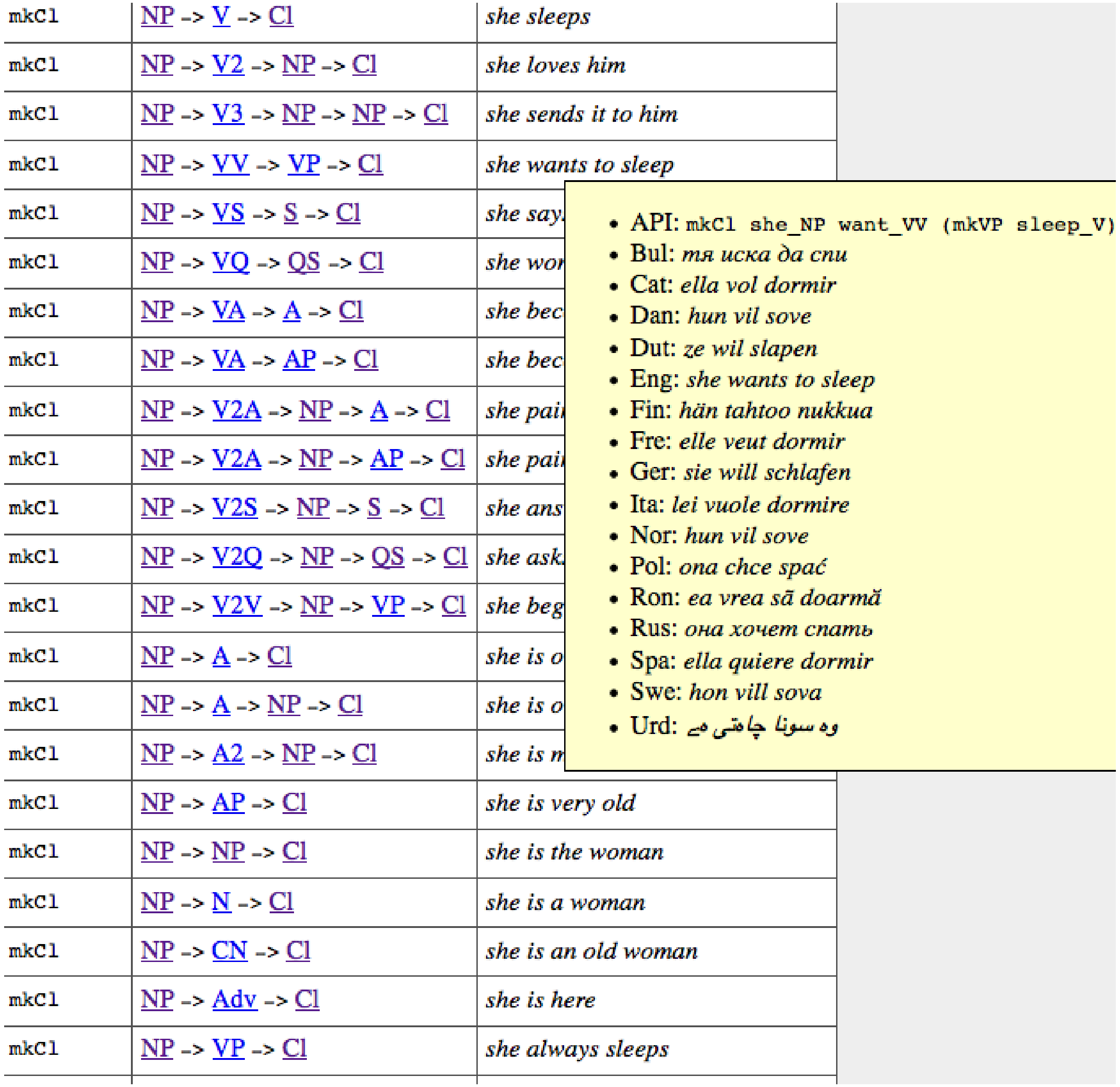
Currently for 16 languages; 3-6 months for a new language.

Complete morphology, comprehensive syntax, lexicon of irregular words.

Common syntax API:

```
  lin Like x y = mkCl x (mkV2 (mkV "like")) y           -- Eng
  lin Like x y = mkCl x (mkV2 (mkV "pitää") elative) y -- Fin
  lin Like x y = mkCl y (mkV2 piacere_V dative) x       -- Ita
  lin Like x y = mkCl x (mkV2 (mkV "tycker") "om") y    -- Swe
```

http://www.grammaticalframework.org/lib/doc/synopsis.html



mkCl	NP -> V -> Cl	*she sleeps*
mkCl	NP -> V2 -> NP -> Cl	*she loves him*
mkCl	NP -> V3 -> NP -> NP -> Cl	*she sends it to him*
mkCl	NP -> VV -> VP -> Cl	*she wants to sleep*
mkCl	NP -> VS -> S -> Cl	*she say*
mkCl	NP -> VQ -> QS -> Cl	*she wor*
mkCl	NP -> VA -> A -> Cl	*she bec*
mkCl	NP -> VA -> AP -> Cl	*she bec*
mkCl	NP -> V2A -> NP -> A -> Cl	*she pai*
mkCl	NP -> V2A -> NP -> AP -> Cl	*she pai*
mkCl	NP -> V2S -> NP -> S -> Cl	*she ans*
mkCl	NP -> V2Q -> NP -> QS -> Cl	*she ask*
mkCl	NP -> V2V -> NP -> VP -> Cl	*she beg*
mkCl	NP -> A -> Cl	*she is o*
mkCl	NP -> A -> NP -> Cl	*she is o*
mkCl	NP -> A2 -> NP -> Cl	*she is m*
mkCl	NP -> AP -> Cl	*she is very old*
mkCl	NP -> NP -> Cl	*she is the woman*
mkCl	NP -> N -> Cl	*she is a woman*
mkCl	NP -> CN -> Cl	*she is an old woman*
mkCl	NP -> Adv -> Cl	*she is here*
mkCl	NP -> VP -> Cl	*she always sleeps*

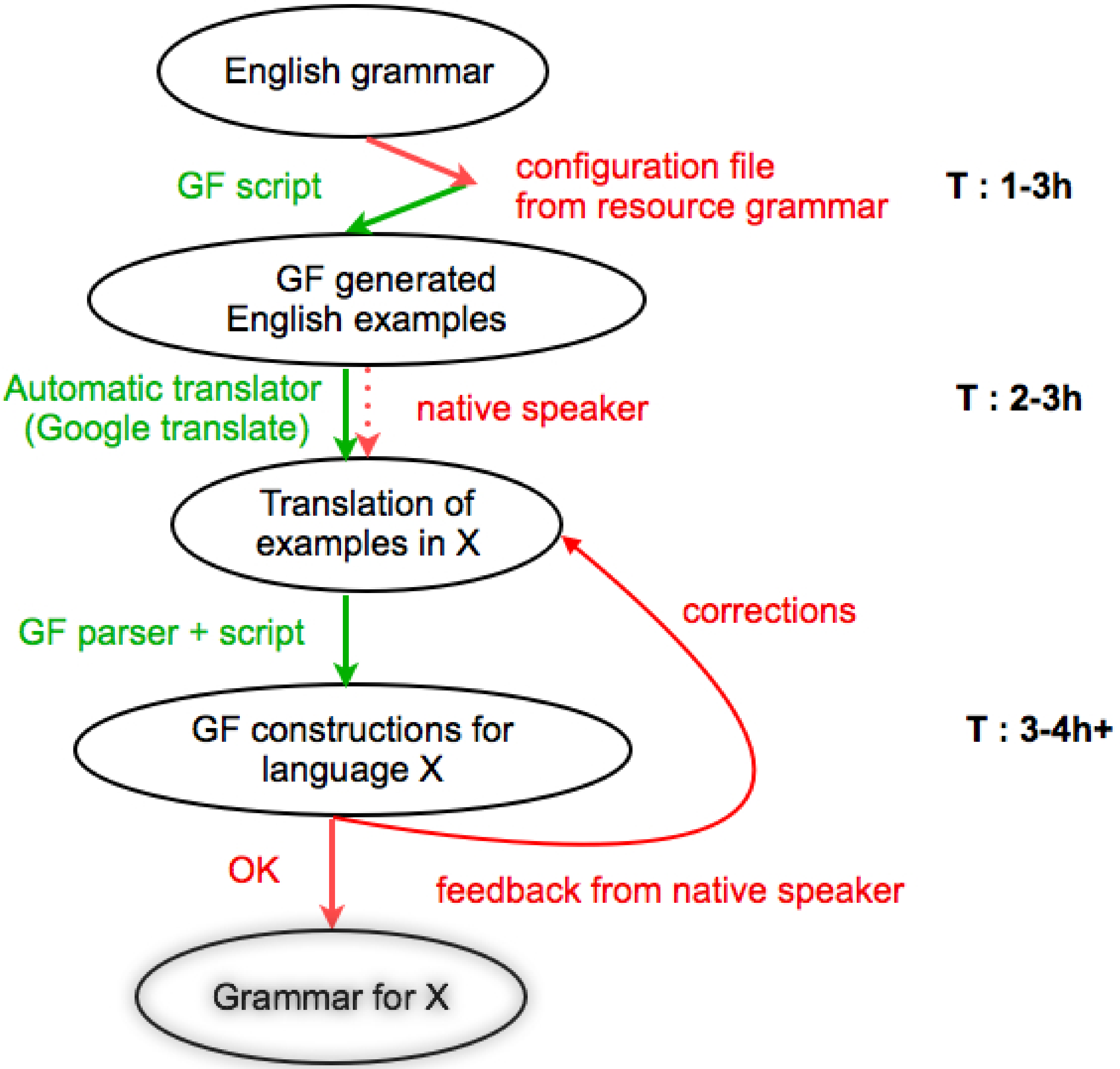
- API: `mkCl she_NP want_VV (mkVP sleep_V)`
- Bul: *тя иска да спи*
- Cat: *ella vol dormir*
- Dan: *hun vil sove*
- Dut: *ze wil slapen*
- Eng: *she wants to sleep*
- Fin: *hän tahtoo nukkua*
- Fre: *elle veut dormir*
- Ger: *sie will schlafen*
- Ita: *lei vuole dormire*
- Nor: *hun vil sove*
- Pol: *ona chce spać*
- Ron: *ea vrea să doarmă*
- Rus: *она хочет спать*
- Spa: *ella quiere dormir*
- Swe: *hon vill sova*
- Urd: وہ سونا چاہتی ہے

Library browsing

Future work

Example-based grammar writing

Abstract syntax	`Like She He`	first grammarian
English example	*she likes him*	first grammarian
German translation	*er gefällt ihr*	human / SMT
resource tree	`mkCl he_NP gefallen_V2 she_NP`	GF parser
concrete syntax rule	`Like x y = mkCl y gefallen_V2 x`	variables renamed



English grammar

GF script

configuration file
from resource grammar

T : 1-3h

GF generated
English examples

Automatic translator
(Google translate)

native speaker

T : 2-3h

Translation of
examples in X

corrections

GF parser + script

GF constructions for
language X

T : 3-4h+

OK

feedback from native speaker

Grammar for X

Effort and cost

Language	Lang skill	GF skill	Infmnt dev	~ test	SMT	SMT impact	RGL chng	Days
Bulgarian	###	###	-	-	-	-	#	##
Catalan	###	###	-	-	-	-	#	#
Danish	-	###	+	+	+	##	#	##
Dutch	-	###	+	+	+	##	#	##
English	##	###	-	+	-	-	-	#
Finnish	###	###	-	-	-	-	#	##
French	##	###	-	+	-	-	#	#
German	#	###	+	+	+	##	##	###
Italian	###	#	-	-	-	-	##	##
Norwegian	#	###	+	-	+	##	#	##
Polish	###	###	+	+	+	#	#	##
Romanian	###	###	-	-	+	#	###	###
Spanish	##	#	-	-	-	-	-	##
Swedish	##	###	-	+	-	-	-	##

MOLTO Phrasebook: 14 languages, 255 concepts (abstract syntax), 300-800 word forms, 27 work days.

Language skill: ### = native, ## = fluent, # = some knowledge

GF skill: ### = expert, ## = 2-3 weeks training, 3 = 1-2 days training

RGL changes: how much changes needed in library

Infmnt: whether informant was used (in development / in testing)

SMT: whether SMT was used in example-based grammar writing

http://www.grammaticalframework.org/examples/phrasebook/doc-phrasebook.html

Dissemination

Presentations

LREC-2010 tutorial, Valletta, May 2010

ACL-2010 demonstration, Uppsala, July 2010

CNL-2010 tutorial, Marettimo, September 2010

TYPES-2010 invited talk, Warsaw, October 2010

SLTC-2010 demonstration, Linköping, October 2010

FreeRBMT-2011 invited talk, Barcelona, January 2011

European Master's invited course, Malta, March 2011

CADE-2011 tutorial and invited talk, Wroclaw, August 2011

GF Summer School 2011, Barcelona, August 2011

Publications

K. Angelov and A. Ranta. Implementing Controlled Languages in GF. N. Fuchs (ed.), CNL-2009 Controlled Natural Languages, LNCS/LNAI 5972, 2010.

O. Caprotti, K. Angelov, R. Enache, Ramona, T. Hallgren, and A. Ranta: The MOLTO Phrasebook. Swedish Language Technology Conference SLTC 2010,

A. Ranta, K. Angelov, and T. Hallgren. Tools for multilingual grammar-based translation on the web. Proceedings of the ACL 2010 System Demonstrations, ACM Digital Library, 2010.

A. Ranta, *Grammatical Framework: Programming with Multilingual Grammars*, CSLI Publications, Stanford, 2011, ISBN-10: 1-57586-626-9 (Paper), 1-57586-627-7 (Cloth).

CSLI Studies in
Computational Linguistics

GRAMMATICAL FRAMEWORK is a programming language designed for writing grammars, which has the capability of addressing several languages in parallel. This thorough introduction demonstrates how to write grammars in Grammatical Framework and use them in applications such as tourist phrasebooks, spoken dialogue systems, and natural language interfaces. The examples and exercises presented here address several languages, and the readers are shown how to look at their own languages from the computational perspective.

Since the book requires no previous knowledge of linguistics, it can be an effective and useful resource for computer scientists and programmers, while introducing linguists to a novel approach to multilingual grammars inspired by the theory of programming languages.

Aarne Ranta is professor of computer science at the University of Gothenburg, Sweden. He is the acting coordinator of the European Union research project MOLTO (Multilingual On-Line Translation), which develops techniques for high-quality translation among fifteen languages.

CSLI
PUBLICATIONS
Center for the Study of
Language and Information
Stanford, California

ISBN-13 978-1-57586-626-0
ISBN-10 1-57586-626-9
9 781575 866260

Aarne Ranta

Grammatical Framework
Programming with Multilingual Grammars

CSLI

CSLI Studies in
Computational Linguistics

Grammatical Framework

Programming with Multilingual Grammars

Aarne Ranta