Multimodal Dialogue System Grammars

Björn Bringert, Peter Ljunglöf,
Aarne Ranta and Robin Cooper

Chalmers University of Technology
and Göteborg University

EU TALK Project, IST-507802

What is Grammatical Framework?

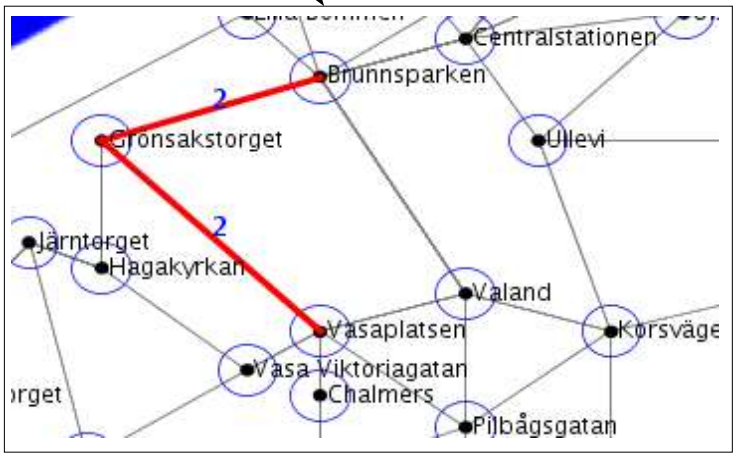
- Grammar formalism
- Based on type theory
- Abstract syntax
- Concrete syntax
  - Linearization functions
- Parsing for free

Parallel Multimodality

- Complete information in each modality:

Route 2 [Brunnsparken, Grönsakstorget, Vasaplatsen]

"Take line two from Brunnsparken to Vasaplatsen"



Parallel Multimodality in GF

- One concrete syntax / modality (like multilinguality):
  - Abstract syntax:

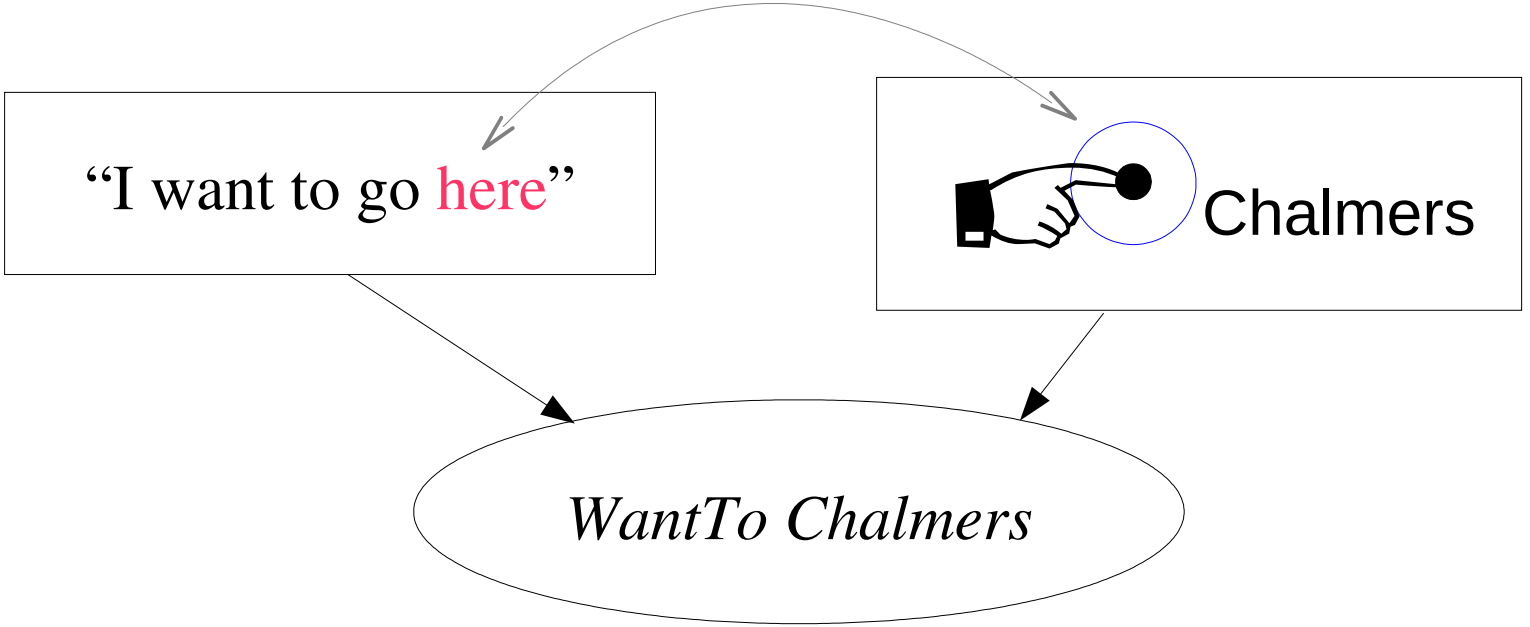
    fun Leg : Line -> Stop -> Stop -> Route
  - English speech:

    lin Leg l f t = { s = "Take" ++ l.s ++ "from" ++ f.s
    ++ "to" ++ t.s }
  - Map drawings:

    lin Leg l f t = { s = "drawEdge(" ++ f.s ++ ","
    ++ t.s ++ "," ++ l.s ++ ")" }

Integrated Multimodality

- Information presented by a combination of modalities:



Integrated Multimodality in GF

- Abstract syntax:

cat Input

cat Place

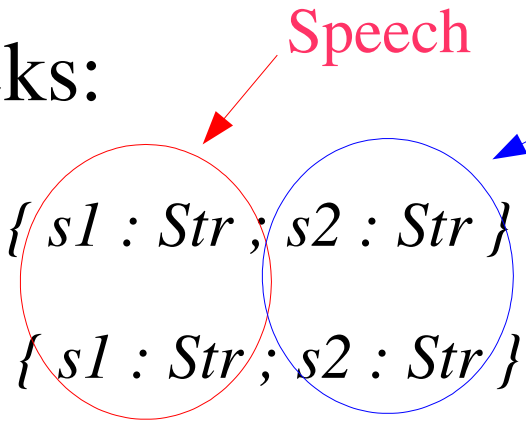
fun GoFromTo : Place -> Place -> Input

fun NamedPlace : String -> Place

fun ClickPlace : Click -> Place

Integrated Multimodality in GF

- English + clicks:

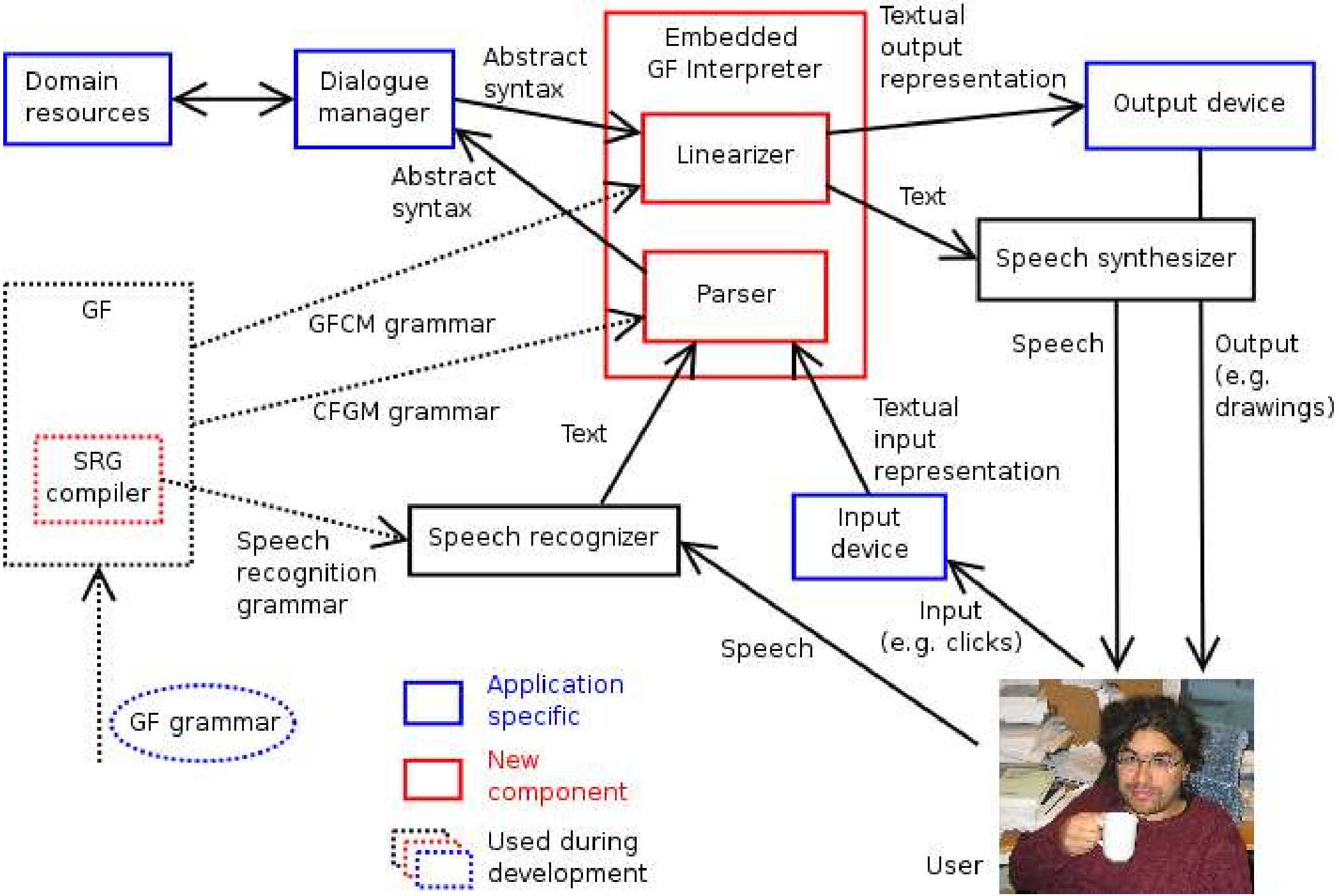



```
lincat Input = { s1 : Str ; s2 : Str }

lincat Place = { s1 : Str ; s2 : Str }

lin GoFromTo x y = {
   s1 = ["I want to go from"] ++ x.s1 ++ "to" ++ y.s1;
   s2 = x.s2 ++ y.s2
}

lin NamedPlace p = { s1 = p.s ; s2 = "" }

lin ClickPlace c = { s1 = "here" ; s2 = c.s }
```

Demo: Göteborg tram map

There is a demo movie at:

http://www.cs.chalmers.se/~bringert/gf/tramdemo.avi


Domain resources
Dialogue manager
Abstract syntax
Embedded GF Interpreter
Linearizer
Parser
Textual output representation
Output device
Abstract syntax
Text
Speech synthesizer
GF
GFCM grammar
CFGM grammar
SRG compiler
Text
Textual input representation
Speech
Output (e.g. drawings)
Speech recognition grammar
Speech recognizer
Input device
Input (e.g. clicks)
Speech
GF grammar
Application specific
New component
Used during development
User

Future Work

- Try more modalities, e.g.:
  - User location (“Switch off the light”)
  - User gestures (“Open that door”)
- Implement dependent type checking in the Embedded GF Interpreter
- Add transfer modules to GF

Conclusions

- Presented a method for writing multimodal grammars
- Embedded GF interpreter makes it easy to use grammars in applications
- Generation of the different grammars avoids problem of keeping grammars in sync