GF Eclipse Plugin

John J. Camilleri

University of Gothenburg

john.j.camilleri@chalmers.se

12 January 2012

The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no. FP7-ICT-247914.
How do we write GF grammars?

Text editor + console

- We have editor modes for Emacs, Gedit, Geany
- Syntax highlighting
- Basic auto-completion

but...

- Separation of writing code and running it
- $GF$-ignorant

Example

Geany
Web IDE

- Good for a *quick start*
- No installation/compilation, always up-to-date
- Storage in the cloud

but...

- Only one module per language
- Impractical for large projects

Example

Web IDE
Why Eclipse?

- Leverage existing IDE features
- Syntax errors & semantic warnings
- Inline documentation
- Context-sensitive suggestions
- Wizards & code snippets
- Code completion, formatting
- Refactoring tools
- Navigate libraries without leaving the IDE
The GF Eclipse Plugin (GFEP)

- Work Package 2: Grammar Developer’s Tools
- Version 1.2.3 (yesterday!)
- Uses Xtext framework
- September 2011 – present
- Still a work-in-progress
Approach

- Layer *on top* of GF, not replace it
- Requires GF already installed on system
- Eclipse **project** concept
  - Regular GF files + metadata files
- Eclipse **builder** concept
  - Compiles your GF code as you write it

Target audience?

- GF newbies
- Application grammar developer
- Seasoned Eclipse users
Getting started

Requirements

• Latest GF from source
• Eclipse 3.7

Installation

• Use Eclipse update site URL:
  http://www.grammaticalframework.org/eclipse/release/
  ⇒ Note about Ubuntu
• Open GF perspective
• Preferences
  • GF path
  • Syntax highlighting
Example

*Foods* grammar

1. New project from existing sources
2. Open/edit some existing files
3. Syntax errors, other warnings
4. Auto-complete
5. Outline view
Example

Launch configurations

- Launch directly into shell
- Interactive or not
Example

- Automatic building
- Unresolved names
- Jump to definition
Example

Let's add a new language! Maybe.. Dutch?

Steps

1. Create new modules with wizard
2. Copy implementation (from English)
3. Code
Example

• Jump to RGL source definition

Browsing the RGL

• Library synopsis
  www.grammaticalframework.org/lib/doc/synopsis.html

• Jumping around source files
Killer feature

Resource Grammar Library browser within IDE
(think Eclipse’s **Type Hierarchy** for Java)
GFEP can:

- leverage useful IDE features
- give you errors & warnings as you type
- help you navigate local & external cross-references
GFEP can:

- leverage useful IDE features
- give you errors & warnings as you type
- help you navigate local & external cross-references

GFEP doesn’t (yet):

- save you as much typing as it could
- have an integrated RGL browser
- provide context-sensitive, type-checked suggestions
- expose any models or APIs
- integrate with anything else
What we hope for GF Eclipse Plugin

- Better parallel-language development tools
- Test suites and treebank management
- Better support for multi-[programming] language projects
- Ontology tool integration
- Whatever else the GF community needs!
We need feedback!
Please report bugs and request features.