Plenary Demo: Isabelle/jEdit — a Generic Frontend for Formal Mathematics

Walther Neuper

IIS Institute Integriert Studieren
Johannes Kepler University Linz
Austria

ThEdu'26 at FLoC
19. July 2026

Outline[1]

1. Demo: Generic Tools of Isabelle/Isar for ...
   - Isabelle/MAWEN with ML
   - Isabelle/jEdit for Haskell
2. Demo: Some features of the generic tools:
   - Elegance and simplicity
   - A glimpse at Isabelle's architecture
3. Summary and Conclusion

[1] For PDF click on  at https://isac.miraheze.org/m/C3B

Outline

Isabelle/MAWEN with ML mathematics engine

Project currently under development, file
`test/main/BridgePIDE/Test_VSCode_Example.thy`

- MAWEN inherits representation of formulas with
  - sub-term structure by mouse hovering
  - syntax errors indicated at correct place
  - linking elements with underlying knowledge
  - 'accessible' by blind people.

- To be shown: integration with suprising little effort

Isabelle/MAWEN with ML mathematics engine

Project currently under development, file
`test/main/BridgePIDE/Test_VSCode_Example.thy`

- MAWEN inherits representation of formulas with
  - sub-term structure by mouse hovering
  - syntax errors indicated at correct place
  - linking elements with underlying knowledge
  - 'accessible' by blind people.
- To be shown: integration with suprising little effort

Outline

Isabelle/jEdit for Haskell

`https://github.com/naproche/naproche`

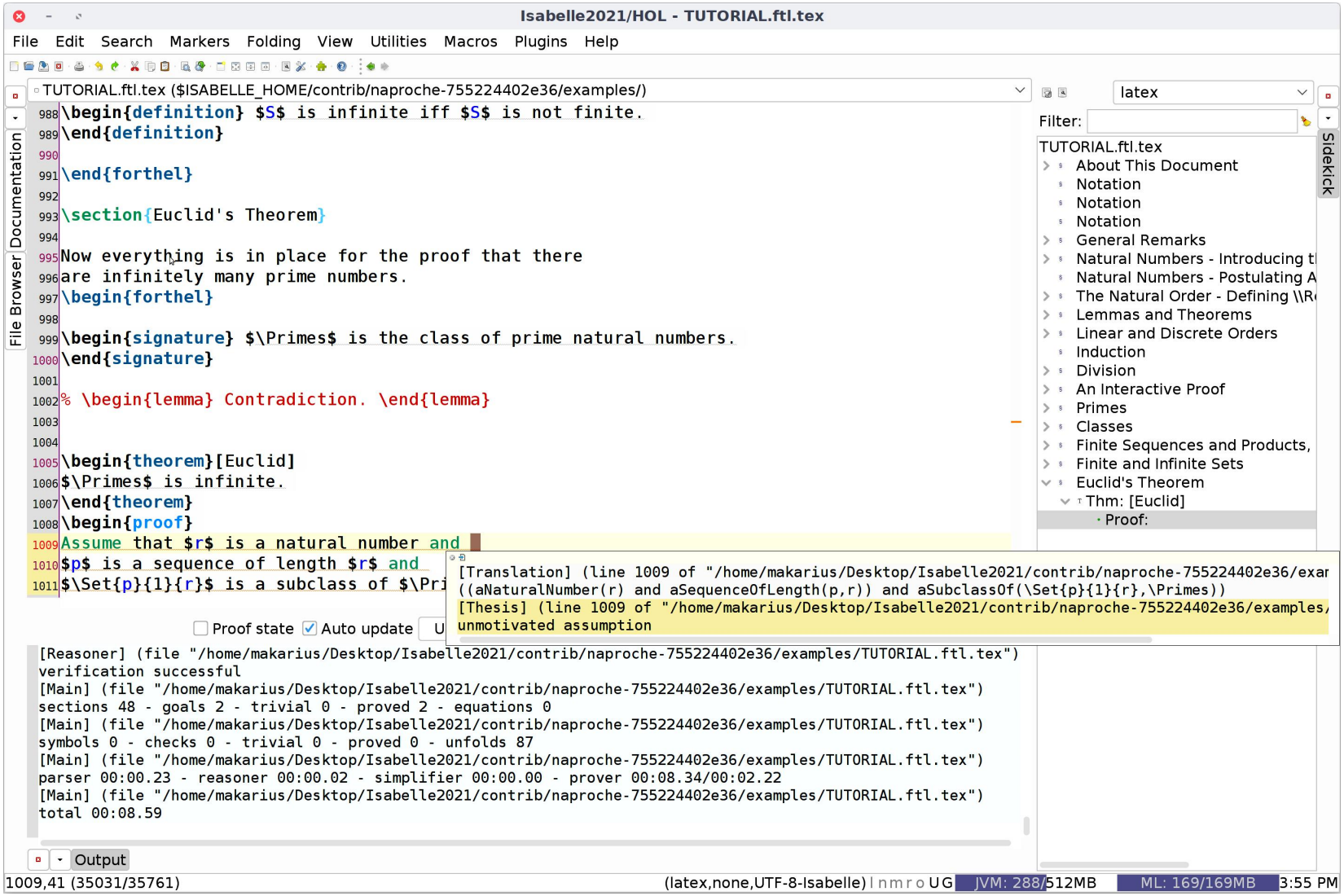
- Proof Checking of Natural Mathematical Documents, with optional support for Isabelle Prover IDE (Isabelle/PIDE – Isabelle/jEdit).
- For End-Users:

  You can get a fully integrated version of Naproche for Linux, macOS and Windows . . .
- For Developers:

  Isabelle/Naproche Prover IDE: Prerequisites, Repository Management, etc

. . . and lots of publications in mathematics and computer science, e.g. `https://www.math.uni-bonn.de/ag/logik/teaching/2024SS/praktikum.shtml`

A glimpse at Naproche with Isabelle/jEdit



Outline

Elegance and simplicity

- Get all features of **formula-representation**:
  Into file
  `test/main/BridgePIDE/Demo_FLoC_26.thy`
  we type . . .
  - . . . two lines into Isabelle/Isar
  - . . . one line into Isabelle/ML
  - . . . and have all the achievements of Isabelle/jEdit auto-magically.
- **System handling** for specific software:
  - e.g. for Isabelle/MAWEN
    `menu 'MAWEN' > 'MAWEN Panel' > 'Browse Knowledge' > 'Examples'`
    - click on an item: student's view (rudimentary)
    - double click: author's view on the definition
  - we have a look at the respective code.
    Files `src/scala/*`

Elegance and simplicity

- Get all features of **formula-representation**:
  Into file
  `test/main/BridgePIDE/Demo_FLoC_26.thy`
  we type . . .
  - . . . two lines into Isabelle/Isar
  - . . . one line into Isabelle/ML
  - . . . and have all the achievements of Isabelle/jEdit auto-magically.
- **System handling** for specific software:
  - e.g. for Isabelle/MAWEN
    `menu 'MAWEN' > 'MAWEN Panel' > 'Browse Knowledge' > 'Examples'`
    - click on an item: student's view (rudimentary)
    - double click: author's view on the definition
  - we have a look at the respective code.
    Files `src/scala/*`

Elegance and simplicity

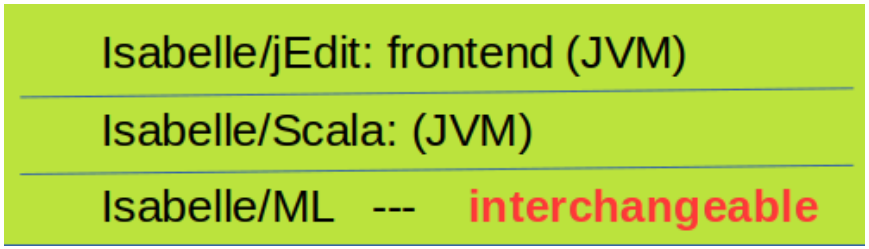
- Get all features of **formula-representation**:
  Into file
  `test/main/BridgePIDE/Demo_FLoC_26.thy`
  we type . . .
  - . . . two lines into Isabelle/Isar
  - . . . one line into Isabelle/ML
  - . . . and have all the achievements of Isabelle/jEdit auto-magically.
- **System handling** for specific software:
  - e.g. for Isabelle/MAWEN
    `menu 'MAWEN' > 'MAWEN Panel' > 'Browse Knowledge' > 'Examples'`
    - click on an item: student's view (rudimentary)
    - double click: author's view on the definition
  - we have a look at the respective code.
    Files `src/scala/*`

Outline

Isabelle's architecture

What is Isabelle [2] ?

- Proof:
  - Isabelle/Isar: Intelligible semi-automated reasoning + **generative tools**
  - Document language: LaTeXtype-setting of proof text
- Programming:
  - Isabelle/Scala: System integration (JVM)



  - Isabelle/ML: Tool implementation (Poly/ML)
- Logic:
  - Isabelle/Pure: Logical framework and bootstrap
  - Isabelle/HOL: Theories and tools for applications

[2] https://sketis.net/2025/invited-talk-at-malinca-kickoff-meeting-in-paris

Summary

- We have shown
  - generic tools of Isabelle/Isar
    - to connect various math SW with Isabelle/jEdit
    - examples for such SW in ML and Haskell
  - the connection auto-magically provides formulas in Isabelle quality:
    - sub-term structure by mouse hovering
    - syntax errors indicated at correct place
    - linking elements with underlying knowledge
    - 'accessible' handling by blind people.
  - tool specific features require coding in Isabelle/Scala

Summary

- We have shown
  - generic tools of Isabelle/Isar
    - to connect various math SW with Isabelle/jEdit
    - examples for such SW in ML and Haskell
  - the connection auto-magically provides formulas in Isabelle quality:
    - sub-term structure by mouse hovering
    - syntax errors indicated at correct place
    - linking elements with underlying knowledge
    - 'accessible' handling by blind people.
  - tool specific features require coding in Isabelle/Scala

Conclusion

- In conclusion, we note that
  - there is no separated API in Isabelle
    (that would hinder further development)
  - rather there are
    - rigorous coding standards ('code is the documentation')
    - software architecture with a high level of abstraction
    - comprehensive documentation
  - Implementing Isabelle/jEdit as a user interface requires
    - a distinguished amount of development work
    - *a secure funding*

- We invite everyone to consider
  whether their software in mathematics or logic
  is worth the effort to create . . .
  - . . . an interface with the hight quality of Isabelle/jEdit
    that is constantly evolving.

Conclusion

- In conclusion, we note that
  - there is no separated API in Isabelle

    (that would hinder further development)
  - rather there are
    - rigorous coding standards ('code is the documentation')
    - software architecture with a high level of abstraction
    - comprehensive documentation
  - Implementing Isabelle/jEdit as a user interface requires
    - a distinguished amount of development work
    - *a secure funding*

- We invite everyone to consider

  whether their software in mathematics or logic
  is worth the effort to create ...
  - ... an interface with the hight quality of Isabelle/jEdit
  that is constantly evolving.

Plenary Demo: *Isabelle/jEdit — a Generic Frontend for Formal Mathematics*

Thank you for attention !

Surprises? Questions?