Semantic Authoring in a Flexiformal Context — Bulk Annotation of Rigorous Documents

Michael Kohlhase, **Jan Frederik Schaefer**

FAU Erlangen-Nürnberg

18th Conference on Intelligent Computer Mathematics
Brasilia, Brazil
October 8, 2025

ALeA Ecosystem

An adaptive learning assistant that

- tracks learner's progress,
- suggests practice problems,
- is powered by semantic annotations

More semantic annotations ⇝ better service

The Cost of Adoption

textbf{Note:} Depth-limited minimax
requires an evaluation for every cut-off state s.
If s is terminal, we use its utility, and otherwise an estimate.

The Cost of Adoption

\textbf{Note:} \sr{depth limit}{ Depth-limited } minimax requires an evaluation for every cut-off state s.
If s is terminal, we use its utility, and otherwise an estimate.

The Cost of Adoption

\usemodule[courses/FAU/AI/course]{search/slides?id-search}

\textbf{Note:} \sr{depth limit}{ Depth-limited } minimax requires an evaluation for every cut-off state s.
If s is terminal, we use its utility, and otherwise an estimate.

The Cost of Adoption

\usemodule[courses/FAU/AI/course]{search/slides?id-search}
\usemodule[courses/FAU/AI/course]{game-play/slides?minimax-algo}

textbf{Note:} \sr{depth limit}{ Depth-limited } \sn{ minimax }
requires an evaluation for every cut-off state s.
If s is terminal, we use its utility, and otherwise an estimate.

The Cost of Adoption

\usemodule[courses/FAU/AI/course]{search/slides?id-search}
\usemodule[courses/FAU/AI/course]{game-play/slides?minimax-algo}

textbf{Note:} \sr{depth limit}{ Depth-limited } \sn{ minimax }
requires an evaluation for every cut-off state s.
If s is terminal, we use its utility, and otherwise an estimate.

This is difficult

- $\geq$ 5 400 concepts in domain model *cannot remember them*
- manual annotation very tedious, especially for novices
- many annotations needed for a lecture *estimate: 10 000*

The Cost of Adoption

\usemodule[courses/FAU/AI/course]{search/slides?id-search}
\usemodule[courses/FAU/AI/course]{game-play/slides?minimax-algo}

textbf{Note:} \sr{depth limit}{ Depth-limited } \sn{ minimax }
requires an evaluation for every cut-off state s.
If s is terminal, we use its utility, and otherwise an estimate.

This is difficult

- $\geq$ 5 400 concepts in domain model *cannot remember them*
- manual annotation very tedious, especially for novices
- many annotations needed for a lecture *estimate: 10 000*

Obvious idea: Have tool support $\rightsquigarrow$ Snify (\sn-ify)

Making a Catalog

\usemodule[courses/FAU/AI/course]{search/slides?id-search}
\usemodule[courses/FAU/AI/course]{game-play/slides?minimax-algo}

textbf{Note:} \sr{depth limit}{Depth-limited} \sn{minimax}
requires an evaluation for every cut-off state s.
If s is terminal, we use its utility, and otherwise an estimate.

- Collect symbol verbalizations from annotations
- Also for other languages (\sr{depth limit}{tiefenbeschränkt})
- Use stemmer to deal with inflections:

  “limit”
“limits”
“limited” $\Big\}\mapsto$ “limit”

Snify

Source-based, incremental, interactive, fine-grained annotation workflow

- Iterate over source files
- Suggest annotations to user

  select annotation instead of remembering and entering it
- Inspired by traditional spell-checkers (e.g. ispell/aspell/...)
- Snify implementation:
  - A workflow experiment
  - Very useful in practice
  - Simple command line interface

Demo

Beyond the Core Idea

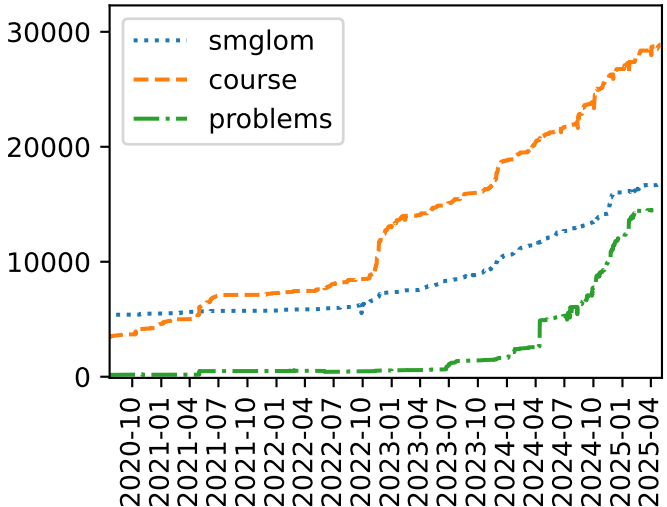
Also needed to make this work:

- Support multiple language
- Add new imports, remove redundant imports
- Only annotate text (not macros, formulae, code, ...)

+ various productivity features:

- Undo/Redo
- Edit in editor
- Select symbol that is not suggested
- Permanently skip word in document/everywhere
- Explain why a symbol is suggested/how it is imported
- Focus mode
- ...

Number of Annotations



Evaluation

Effect:

- Productivity gain over sTeX IDE: $3\times - 13\times$
- Higher annotation density
- Used by most sTeX annotators
- Has been used for estimated 10 000 annotations, possibly 20 000

Limitations:

- Catalog must be reasonably complete for document language
- Only words in the catalog are offered $\rightsquigarrow$ less awareness of gaps in domain model?
- Does not support annotation while writing
- Only supports annotation of symbol references in text

no formulae, no metadata, . . .

Side Effect: Debugging the Domain Model

Different perspective on domain model with focus on what we care about right now.
Snify helps detect e.g.:

- Duplicate symbols
- Gaps in the domain model
- Unnecessary dependencies

Side Effect: Debugging the Domain Model

Different perspective on domain model with focus on what we care about right now.
Snify helps detect e.g.:

- Duplicate symbols
- Gaps in the domain model
- Unnecessary dependencies

```
[0] × smglom/logic mod?evaluation?evaluate
[1] ✓ courses/FAU/AI/course game-play/slides?evaluation-function?evaluation function
[2] × smglom/logic mod?evaluation?evaluation
[3] × smglom/computing mod?program-expression?evaluation
[4] × smglom/computing mod?program-expression?evaluate
[5] × mkohlhase/talks flexiforms/slides?mathdoc-structure?evaluation
[6] × smglom/education mod?assessment?assess
```

Semantic Authoring

	Informal Authoring	**Semantic Authoring**	**Formal Authoring**
Example	informal LaTeX	sTeX	computer code

Semantic Authoring

	Informal Authoring	**Semantic Authoring**	**Formal Authoring**
Example	informal LaTeX	sTeX	computer code
People	mathematicians	we	formalizers

Semantic Authoring

	Informal Authoring	**Semantic Authoring**	**Formal Authoring**
Example	informal LaTeX	sTeX	computer code
People	mathematicians	we	formalizers
Dom. model	in head/literature	glossary	libraries

Semantic Authoring

	Informal Authoring	**Semantic Authoring**	**Formal Authoring**
Example	informal LaTeX	sTeX	computer code
People	mathematicians	we	formalizers
Dom. model	in head/literature	glossary	libraries
Reuse	limited	more desirable & possible	all the time

Semantic Authoring

	Informal Authoring	**Semantic Authoring**	**Formal Authoring**
Example	informal LaTeX	sTeX	computer code
People	mathematicians	we	formalizers
Dom. model	in head/literature	glossary	libraries
Reuse	limited	more desirable & possible	all the time
Tools	spell/grammar checker	**?**	IDE (refactoring, ...)

Semantic Authoring

	Informal Authoring	**Semantic Authoring**	**Formal Authoring**
Example	informal LaTeX	sTeX	computer code
People	mathematicians	we	formalizers
Dom. model	in head/literature	glossary	libraries
Reuse	limited	more desirable & possible	all the time
Tools	spell/grammar checker	**?**	IDE (refactoring, ...)

Semantic authoring tool support:

- spell/grammar checking — *like informal authoring*
- IDE functionality (e.g. refactoring symbol names) — *like formal authoring*
- need tools that combine both aspects — *like Snify*

Design Space of Semantic Authoring

There are multiple dimensions to consider:

- Text vs formulae vs metadata
- Incremental vs holistic
- During writing vs afterwards

or combination?

These decisions affect cognitive load

Design Space of Semantic Authoring

There are multiple dimensions to consider:

- Text vs formulae vs metadata
- Incremental vs holistic
- During writing vs afterwards

or combination?

These decisions affect cognitive load

Conclusion

- Semantic authoring shares aspects of formal and informal authoring
- We need tools that combine both aspects
- Snify case study
  - Maintains verbalization catalog
  - Uses simple symbolic techniques
  - Is very effective

no need for LLMs
already exists for Word

```
                                    /tmp/examplefile.tex
  4 \usemodule[courses/FAU/AI/course]{game-play/slides?evaluation-function}
  5
  6 % ...
  7
  8 \textbf{Note:} \sr{depth limit}{Depth-limited} minimax requires an
  9 evaluation for every cut-off state $s$.
 10 If $s$ is terminal, we use its utility, and otherwise an estimate.
 11
 12 \end{document}

Commands: enter h (help) to see all available commands
  [h]elp
  [q]uit
  [0] ✓ courses/FAU/AI/course game-play/slides?minimax-algo?minimax
  [s]kip once
>>>
```