Proxima: A generic XML editor

Martijn Schrage, Rui Guerra, Johan Jeuring, Doaitse Swierstra, and Lambert Meertens

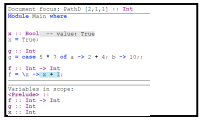
XML and structure editing

A generic XML editor is an editor for a whole class of document types, rather than a specific type. The editor is specialized with a type definition and a number of specifications (style sheet, edit sheet, etc.)

Building an instance of a generic editor requires only a fraction of the effort that is needed to build a custom-made editor.

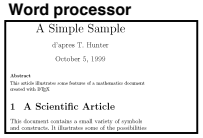
A few sample applications:

Haskell source editor



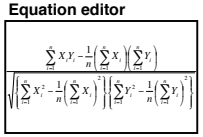
- Free text editing
- Type info in source
- Optional automatic layout
- Navigation (eg. jump to declaration)
- Hide function definitions
- Smart editing: - rename within scope
  - move functions

Word processor



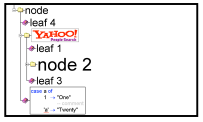
- Latex quality
- "Underwater" view
- Automatic table of contents and bibliography
- Navigation (eg. jump to figure)
- Edit in table of contents

Equation editor



- Textual input of expressions
- Domain specific transformations
- Drag and drop

Tree browser



- Non-primitive, hence customizable
- Drag and drop
- Navigation

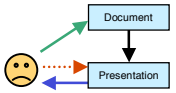
Tax form

- Widgets: buttons, list-boxes, etc.
- Presentation structure depends on input
- Computed values in presentation

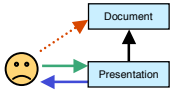
Actual editors may combine features from any of these examples.

Structure editors

Syntax-directed: (includes XML editors)
- *Document* is main data structure
- Presentation is computed from document
- eg. Synthesizer Generator, LRC, XMetal
- PRO: supports document editing: eg. insert **if** statement, move subsection
- PRO: computations & presentation
- CON: restrictive, no free editing in presentation



Syntax-recognizing:
- *Presentation* is main data structure
- Document is computed from presentation
- eg. Pan, Ensemble, Visual Studio
- PRO: supports presentation editing: eg. type keywords, change layout
- PRO: editing freedom
- CON: limited document editing, and simple presentations (text-only)



Hybrid: term is used inconsistently, usually either syntax-directed or syntax-recognizing

Requirements

No existing editor is able to handle all of the examples!

We believe the reason is that no editor meets all of the following requirements:

Genericity
Document types can be specified with a formalism similar to EBNF grammars, DTDs or XML Schemas.

Support for computations over the document
A formalism powerful enough for type-checking, specifying derived structures (TOC), and spread-sheet behaviour (tax form).

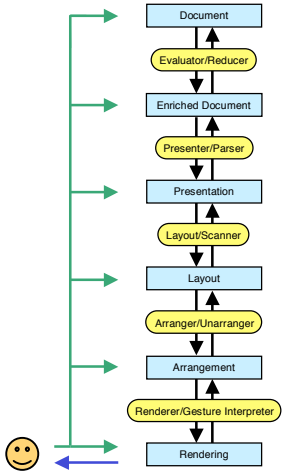
Graphical presentations with powerful mapping formalism
For formatting paragraphs and mathematics formulas, Style sheets must be transparent and easy to reuse.

Editing on all levels
Not only document and presentation editing, but also on derived structures.

Modeless editing
User can mix edit operations on different levels, without explicitly having to switch editor modes.

Support for local state
Presentation and other levels may have state that is not part of the document. (eg. tree browser expansion state, whitespace, color)

Proxima architecture



Internal document tree

Document + derived values (eg. numbers, types, table of contents)

Logical presentation: rows, columns, matrices, formatters, strings, lines and images.

Presentation + whitespace

Formatted presentation with absolute positions & sizes

Bitmap

Proxima features

Proxima will be able to handle all 5 examples.

Features that allow Proxima to meet the requirements:

- AG formalism for specifying computations and presentations
- Xprez declarative presentation language
- A layered architecture described with Haskell combinators
- Bidirectional mappings between document and presentation
- Local state on several levels of the presentation process
- Declarative specification languages with strong abstraction mechanisms for specifying mappings between levels

•Prototype written entirely in Haskell (15.000 lines of code)

Contact information

Martijn Schrage
e-mail: **martijn@cs.uu.nl**
www: **http://www.cs.uu.nl/research/projects/proxima**

Prototype: Haskell source editor