

Proxima 2.0: Web-based generic editing

<http://www.cs.uu.nl/wiki/Proxima/WebHome>

Martijn Schrage, Lambert Meertens, and Doaitse Swierstra

Oblomov Systems
Utrecht

Department of Computer Science
Utrecht University

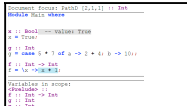
Generic structure editing

A generic structure 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 with corresponding requirements:

Haskell source editor



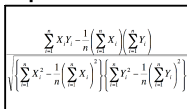
- Free text editing
- Type info in source
- Optional automatic layout
- Navigation (e.g. jump to declaration)
- Hide function definitions
- Smart editing: - rename within scope
- move functions

Word processor



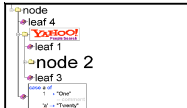
- Latex quality
- "Underwater" view
- Automatic table of contents and bibliography
- Navigation (e.g. jump to figure)
- Edit in table of contents

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.

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 behavior (tax form).

Graphical presentations with powerful mapping formalism

For formatting paragraphs and mathematical formulas.
Style sheets must be transparent and easy to reuse.

Editing on all levels

Not only document and presentation editing, also on derived structures.

Modeless editing

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

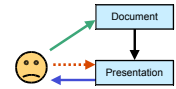
Support for extra state

Presentation and other levels may have state that is not part of the document. (e.g. tree browser expansion state, whitespace, color)

Structure editors

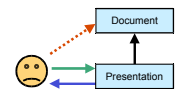
Syntax-directed: (includes XML editors)

- *Document* is main data structure
- Presentation is computed from document
- e.g. Synthesizer Generator, LRC, XMetal
- PRO: supports document editing:
e.g. insert **if** statement, move subsection
- PRO: computations & presentation
- CON: restrictive, no free editing in presentation



Syntax-recognizing:

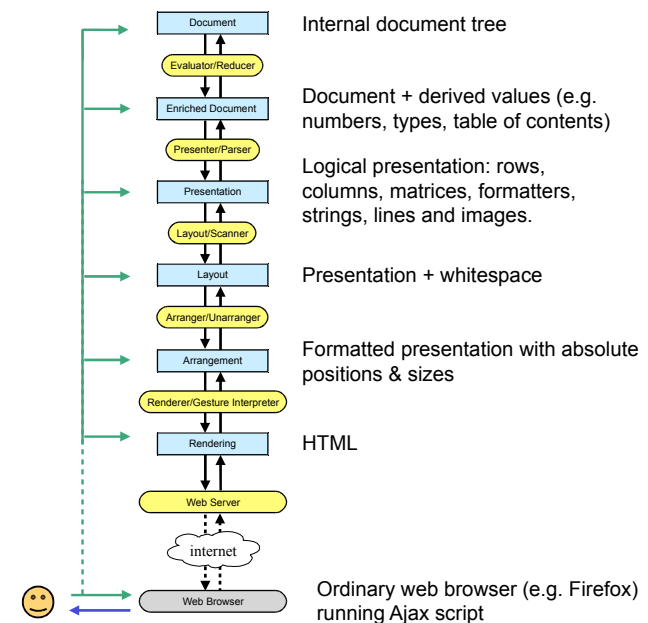
- *Presentation* is main data structure
- Document is computed from presentation
- e.g. Pan, Ensemble, Visual Studio
- PRO: supports presentation editing:
e.g. type keywords, change layout
- PRO: editing freedom
- CON: limited document editing, and simple presentations (text-only)



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



Proxima 2.0 architecture



Proxima 2.0 features

Proxima 2.0 will be able to handle all 5 examples, and support editing in any browser, not requiring any installation on the client machine.

Features that allow Proxima to meet the requirements:

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

The web server incrementally sends the rendering to an Ajax client that displays it and catches edit events, which are sent back to the server.

Prototype written in Haskell (20 000 lines of code) with Ajax client (1 000 lines).

The Proxima 2.0 project is supported by a grant from NLnet Foundation.