A Math-Canvas for GNOME

Luca Padovani

Department of Computer Science

University of Bologna

Member of the W3C Math Interest Group http://www.w3.org/Math

Summary

- MathML

  brief introdeuction to the markup language for mathematics

- Using `GtkMathView`

  create GTK+ applications with views for mathematics, embedding `GtkMathView`

- `GtkMathView` internals

  porting and adapting

- Future developments

MathML

MathML Presentation: example

MathML Presentation: example

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
```

$$\lim_{x \to 0} \frac{\sin x}{x} = 25$$

```
</math>
```

MathML Presentation: example

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>






















  </mrow>
</math>
```

$$\lim_{x \to 0} \frac{\sin x}{x} = 25$$

MathML Presentation: example

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mrow>
```

$$\lim_{x\to 0} \frac{\sin x}{x} = 25$$

```
    </mrow>
    <mo> = </mo>
    <mn> 25 </mn>
  </mrow>
</math>
```

MathML Presentation: example

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mrow>
      <munder>
        <mo> lim </mo>
        <mrow>



        </mrow>
      </munder>
      <mfrac>
        <mrow>



        </mrow>
        <mi> x </mi>
      </mfrac>
    </mrow>
    <mo> = </mo>
    <mn> 25 </mn>
  </mrow>
</math>
```

$$\lim_{x\to 0}\frac{\sin x}{x} = 25$$

MathML Presentation: example

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mrow>
      <munder>
        <mo> lim </mo>
        <mrow>
          <mi> x </mi>
          <mo> &RightArrow; </mo>
          <mn> 0 </mn>
        </mrow>
      </munder>
      <mfrac>
        <mrow>
          <mi> sin </mi>
          <mo> &ApplyFunction; </mo>
          <mi> x </mi>
        </mrow>
        <mi> x </mi>
      </mfrac>
    </mrow>
    <mo> = </mo>
    <mn> 25 </mn>
  </mrow>
</math>
```

$$\lim_{x \to 0} \frac{\sin x}{x} = 25$$

MathML Presentation: example

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mrow>
      <munder>
        <mo> lim </mo>
        <mrow>
          <mi> x </mi>
          <mo> &RightArrow; </mo>
          <mn> 0 </mn>
        </mrow>
      </munder>
      <mfrac>
        <mrow>
          <mi> sin </mi>
          <mo> &ApplyFunction; </mo>
          <mi> x </mi>
        </mrow>
        <mi> x </mi>
      </mfrac>
    </mrow>
    <mo> = </mo>
    <mn> 25 </mn>
  </mrow>
</math>
```

$$\lim_{x\to 0}\frac{\sin x}{x} = 25$$

MathML Presentation Summary

- tokens (`mi`, `mo`, `mn`)
- general layout schemata (`mfrac`, `msqrt`)
- scripts and limits (`msub`, `msup`, `munder`, `mover`)
- tables and alignment (`mtable`, `mtr`, `mtd`)
- style and attribute inheritance (`mstyle`)
- “live” expressions (`maction`)

MathML Presentation Summary

- tokens (`mi`, `mo`, `mn`)
- general layout schemata (`mfrac`, `msqrt`)
- scripts and limits (`msub`, `msup`, `munder`, `mover`)
- tables and alignment (`mtable`, `mtr`, `mtd`)
- style and attribute inheritance (`mstyle`)
- “live” expressions (`maction`)

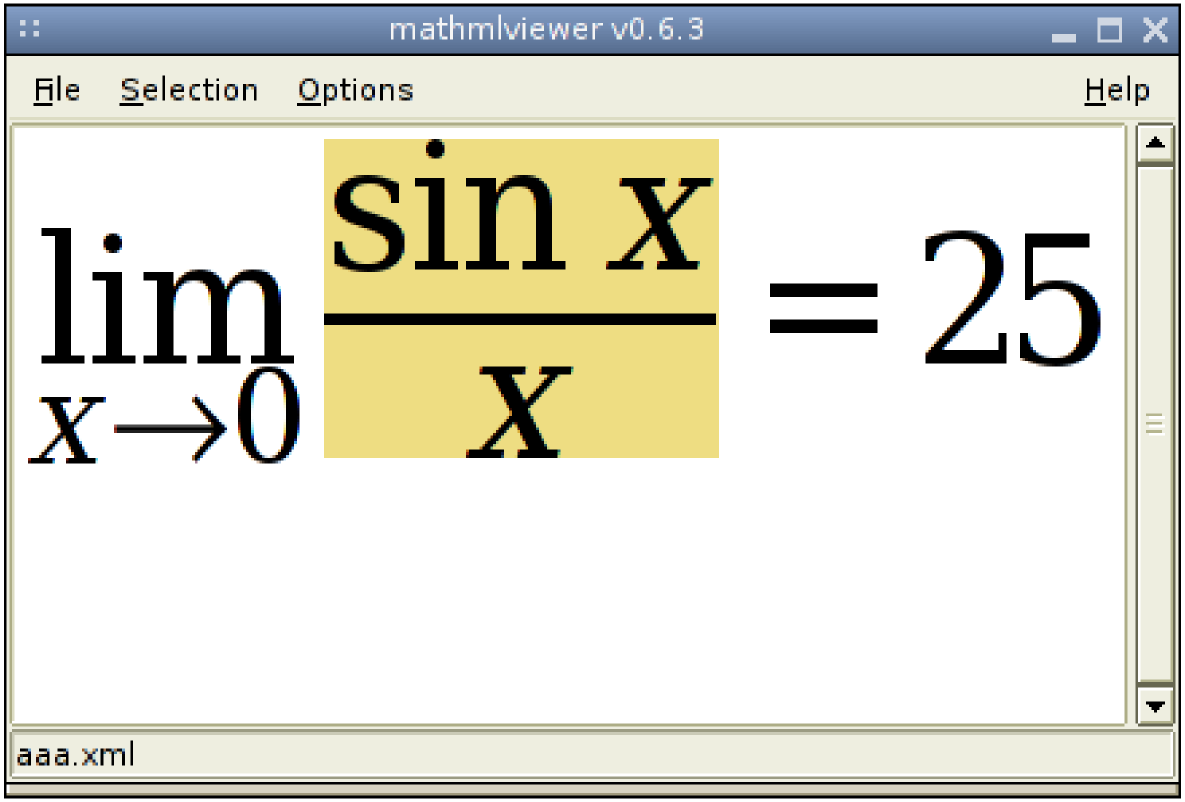
Semantics in presentation elements:

- refine formatting, higher quality
- “meaningful” presentation (conversions)

Usage

Basic Use

- Independent component for displaying MathML markup;
- `mathmlviewer` application.



$$\lim_{x\to 0}\frac{\sin x}{x} = 25$$

aaa.xml

Basic Use

1. `GtkMathView* view = gtk_math_view_new();`

Basic Use

1. `GtkMathView* view = gtk_math_view_new();`

2. `gtk_math_view_load_uri(view, "http://my mathml document");`

Basic Use

1. `GtkMathView* view = gtk_math_view_new();`
2. `gtk_math_view_load_uri(view, "http://my mathml document");`

or... `gtk_math_view_load_document(view, doc);`

or... `gtk_math_view_load_root(view, node);`

or... `gtk_math_view_load_buffer(view, "<math xmlns=...");`

Basic Use

1. `GtkMathView* view = gtk_math_view_new();`

2. `gtk_math_view_load_uri(view, "http://my mathml document");`

or... `gtk_math_view_load_document(view, doc);`

or... `gtk_math_view_load_root(view, node);`

or... `gtk_math_view_load_buffer(view, "<math xmlns=...");`

3. enjoy

Basic Use

1. `GtkMathView* view = gtk_math_view_new();`
2. `gtk_math_view_load_uri(view, "http://my mathml document");`

   or... `gtk_math_view_load_document(view, doc);`

   or... `gtk_math_view_load_root(view, node);`

   or... `gtk_math_view_load_buffer(view, "<math xmlns=...");`
3. enjoy
4. `gtk_math_view_unload(view);`

Cursors and Model Identifiers

Cursors:

- plain text: offset
- structured document: node id + offset

Cursors and Model Identifiers

Cursors:

- plain text: offset
- structured document: node id + offset

Depending on how the source document model is represented, `GtkMathViewModelId` is:

- `GdomeElement*` for `Gdome2`;
- `xmlElement*` for `libxml2`;
- `void*` for custom document models.

Clicking

```
void (*click)(GtkMathView*, const GtkMathViewModelEvent*);

typedef struct _GtkMathViewModelEvent {
  GtkMathViewModelId id;
  gint x;
  gint y;
  gint state;
} GtkMathViewModelEvent;
```

- application-specific actions
- MathML actions
- context-sensitive popup menus

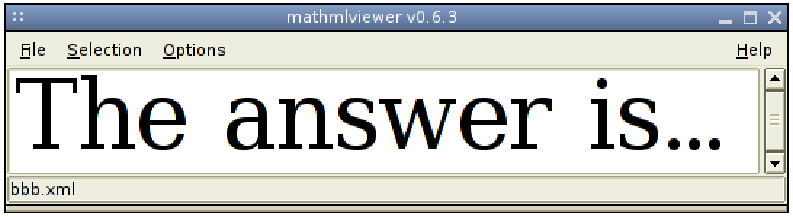
Activating Actions

```
<maction actiontype="toggle" selection="1">
  <mtext>The answer is...</mtext>
  <msqrt> <mn mathcolor="red">2</mn> </msqrt>
</maction>
```

Activating Actions

```
<maction actiontype="toggle" selection="1">
  <mtext>The answer is...</mtext>
  <msqrt> <mn mathcolor="red">2</mn> </msqrt>
</maction>
```

⇒

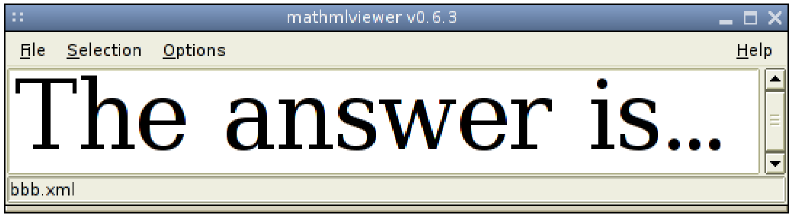


The answer is...

Activating Actions

```
<maction actiontype="toggle" selection="1">
  <mtext>The answer is...</mtext>
  <msqrt> <mn mathcolor="red">2</mn> </msqrt>
</maction>
```

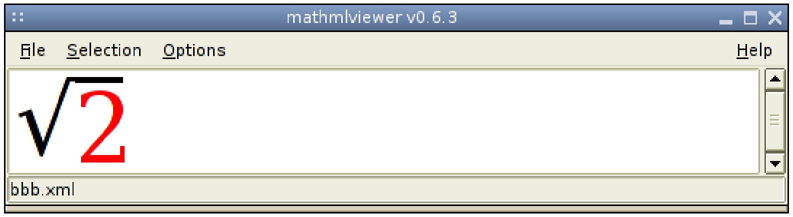
⇒



The answer is...

bbb.xml

click ⇒



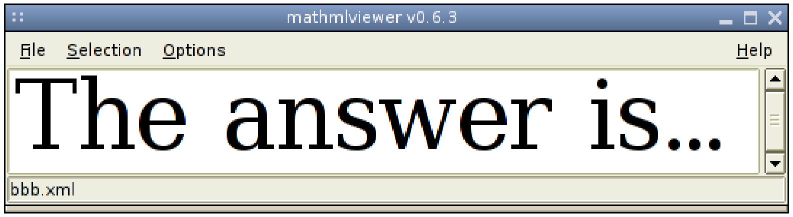
$\sqrt{2}$

bbb.xml

Activating Actions

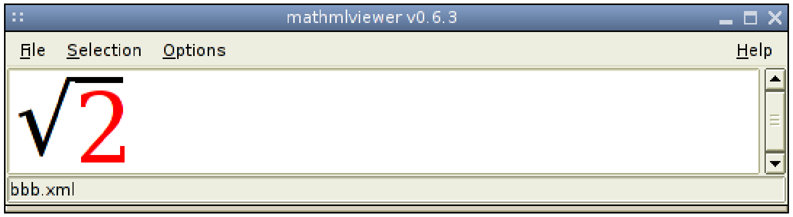
```
<maction actiontype="toggle" selection="1">
  <mtext>The answer is...</mtext>
  <msqrt> <mn mathcolor="red">2</mn> </msqrt>
</maction>
```

⇒



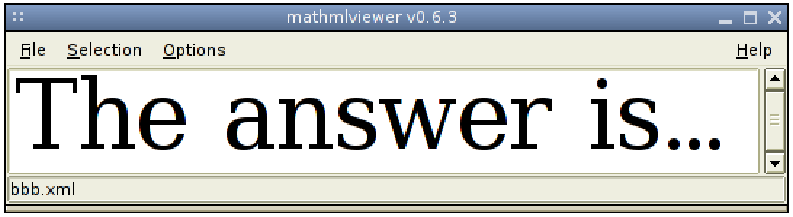
The answer is...

click ⇒



$\sqrt{2}$

click ⇒



The answer is...

Activating Actions

```
static void
click(GtkMathView* math_view, const GtkMathViewModelEvent* event)
{
  if (event->id != NULL)
    {
      GtkMathViewModelId action = find_action(event->id);
      if (action != NULL)
        {
          gtk_math_view_freeze(math_view);
          action_toggle(action);
          gtk_math_view_thaw(math_view);
        }
    }
}
```

Activating Actions

```
static void
click(GtkMathView* math_view, const GtkMathViewModelEvent* event)
{
  if (event->id != NULL)
    {
      GtkMathViewModelId action = find_action(event->id);
      if (action != NULL)
        {
          gtk_math_view_freeze(math_view);
          action_toggle(action);
          gtk_math_view_thaw(math_view);
        }
    }
}
```

Activating Actions

```
static void
click(GtkMathView* math_view, const GtkMathViewModelEvent* event
{
  if (event->id != NULL)
    {
      GtkMathViewModelId action = find_action(event->id);
      if (action != NULL)
        {
          gtk_math_view_freeze(math_view);
          action_toggle(action);
          gtk_math_view_thaw(math_view);
        }
    }
}
```

Activating Actions

```
static void
click(GtkMathView* math_view, const GtkMathViewModelEvent* event)
{
  if (event->id != NULL)
    {
      GtkMathViewModelId action = find_action(event->id);
      if (action != NULL)
        {
          gtk_math_view_freeze(math_view);
          action_toggle(action);
          gtk_math_view_thaw(math_view);
        }
    }
}
```

Selecting

Structured document ⇒ several options:

- structural selection
- linear selection (document order)
- linear selection (layout order)

Selecting

Structured document ⇒ several options:

- structural selection
- linear selection (document order)
- linear selection (layout order)

Moreover, desirable to have multiple selections.

Selecting

Select signal sequences:

$$select\_begin\ \ select\_over^{*}\ \ (select\_end \mid select\_abort)$$

Signals:

```
void (*select_...)(GtkMathView*, const GtkMathViewModelEvent*);
void (*select_abort)(GtkMathView*);
```

Selecting

Select signal sequences:

$$select\_begin\ select\_over^{*}\ (select\_end \mid select\_abort)$$

Signals:

```
void (*select_...)(GtkMathView*, const GtkMathViewModelEvent*);
void (*select_abort)(GtkMathView*);
```

Policy and mechanism are separated.

Selecting

Select signal sequences:

$$select\_begin\ \ select\_over^{*}\ \ (select\_end \mid select\_abort)$$

Signals:

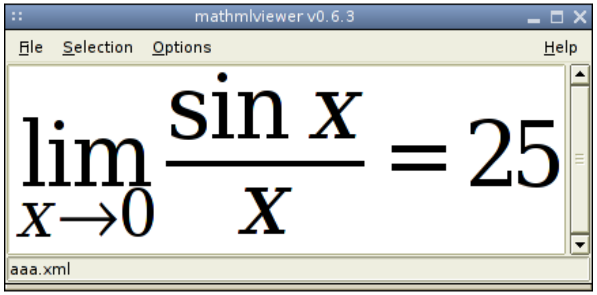
```
void (*select_...)(GtkMathView*, const GtkMathViewModelEvent*);
void (*select_abort)(GtkMathView*);
```

Policy and mechanism are separated.

Methods for selections:

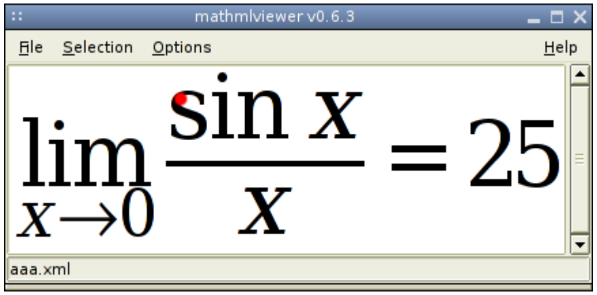
```
void gtk_math_view_select(GtkMathView*, GtkMathViewModelId);
void gtk_math_view_unselect(GtkMathView*, GtkMathViewModelId);
gboolean gtk_math_view_is_selected(GtkMathView*,
                                   GtkMathViewModelId);
```

Structural selection: example



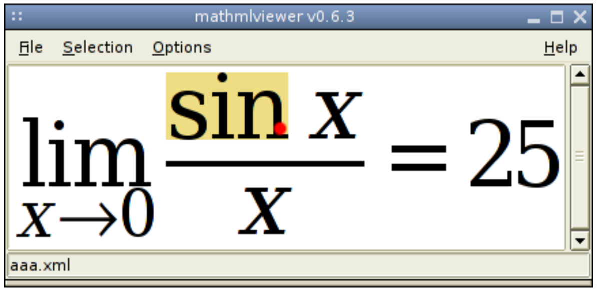
$$\lim_{x\to 0}\frac{\sin x}{x}=25$$

1



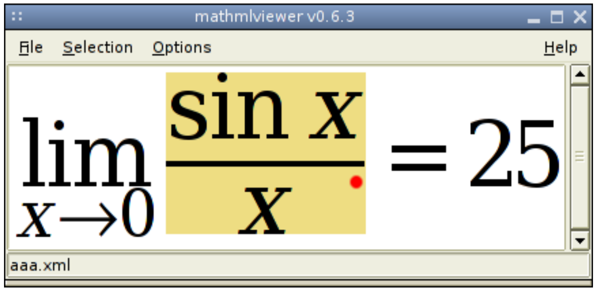
$$\lim_{x\to 0}\frac{\sin x}{x}=25$$

2



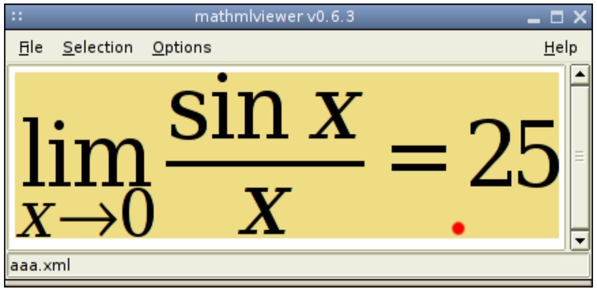
$$\lim_{x\to 0}\frac{\sin x}{x}=25$$

3



$$\lim_{x\to 0}\frac{\sin x}{x}=25$$

4



$$\lim_{x\to 0}\frac{\sin x}{x}=25$$

5

Structural selection: implementation

```
static void
select_over(GtkMathView* math_view,
            const GtkMathViewModelEvent* event)
{
  if (first != NULL && event->id != NULL)
    {
      gtk_math_view_freeze(math_view);
      if (root != NULL)
        gtk_math_view_unselect(math_view, root);
      root = find_common_ancestor(first, event->id);
      if (root != NULL)
        gtk_math_view_select(math_view, root);
      gtk_math_view_thaw(math_view);
    }
}
```

Structural selection: implementation

```
static void
select_over(GtkMathView* math_view,
            const GtkMathViewModelEvent* event)
{
  if (first != NULL && event->id != NULL)
    {
      gtk_math_view_freeze(math_view);
      if (root != NULL)
        gtk_math_view_unselect(math_view, root);
      root = find_common_ancestor(first, event->id);
      if (root != NULL)
        gtk_math_view_select(math_view, root);
      gtk_math_view_thaw(math_view);
    }
}
```

Structural selection: implementation

```
static void
select_over(GtkMathView* math_view,
            const GtkMathViewModelEvent* event)
{
  if (first != NULL && event->id != NULL)
    {
      gtk_math_view_freeze(math_view);
      if (root != NULL)
        gtk_math_view_unselect(math_view, root);
      root = find_common_ancestor(first, event->id);
      if (root != NULL)
        gtk_math_view_select(math_view, root);
      gtk_math_view_thaw(math_view);
    }
}
```

Structural selection: implementation

```
static void
select_over(GtkMathView* math_view,
            const GtkMathViewModelEvent* event)
{
  if (first != NULL && event->id != NULL)
    {
      gtk_math_view_freeze(math_view);
      if (root != NULL)
        gtk_math_view_unselect(math_view, root);
      root = find_common_ancestor(first, event->id);
      if (root != NULL)
        gtk_math_view_select(math_view, root);
      gtk_math_view_thaw(math_view);
    }
}
```

Structural selection: implementation

```
static void
select_over(GtkMathView* math_view,
            const GtkMathViewModelEvent* event)
{
  if (first != NULL && event->id != NULL)
    {
      gtk_math_view_freeze(math_view);
      if (root != NULL)
        gtk_math_view_unselect(math_view, root);
      root = find_common_ancestor(first, event->id);
      if (root != NULL)
        gtk_math_view_select(math_view, root);
      gtk_math_view_thaw(math_view);
    }
}
```

Structural selection: implementation

```
static void
select_over(GtkMathView* math_view,
            const GtkMathViewModelEvent* event)
{
  if (first != NULL && event->id != NULL)
    {
      gtk_math_view_freeze(math_view);
      if (root != NULL)
        gtk_math_view_unselect(math_view, root);
      root = find_common_ancestor(first, event->id);
      if (root != NULL)
        gtk_math_view_select(math_view, root);
      gtk_math_view_thaw(math_view);
    }
}
```

Structural selection: implementation

```
static void
select_over(GtkMathView* math_view,
            const GtkMathViewModelEvent* event)
{
  if (first != NULL && event->id != NULL)
    {
      gtk_math_view_freeze(math_view);
      if (root != NULL)
        gtk_math_view_unselect(math_view, root);
      root = find_common_ancestor(first, event->id);
      if (root != NULL)
        gtk_math_view_select(math_view, root);
      gtk_math_view_thaw(math_view);
    }
}
```

Editing

Limited support:

- the widget can draw focus and/or caret;

Editing

Limited support:

- the widget can draw focus and/or caret;
- the widget listens to change notifications on the model;

Editing

Limited support:

- the widget can draw focus and/or caret;
- the widget listens to change notifications on the model;
- the widget supports incremental formatting.

Editing

Limited support:

- the widget can draw focus and/or caret;
- the widget listens to change notifications on the model;
- the widget supports incremental formatting.

No editing API:

```
void gtk_math_view_structure_changed(GtkMathView*,
                                     GtkMathViewModelId);
void gtk_math_view_attribute_changed(GtkMathView*,
                                     GtkMathViewModelId,
                                     GtkMathViewModelString);
```

EdiTeX

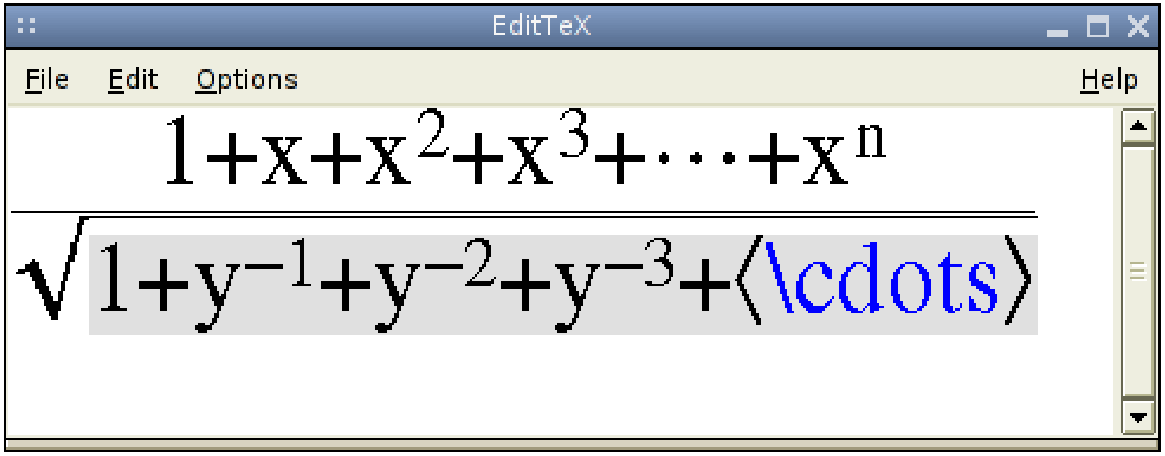
Syntax-directed editing using TeX syntax:

```
\frac{1+x+x^2+x^3+\cdots+x^n}
{\sqrt{1+y^{-1}+y^{-2}+y^{-3}+\cdots
```

EdiTeX

Syntax-directed editing using TeX syntax:

```
\frac{1+x+x^2+x^3+\cdots+x^n}
{\sqrt{1+y^{-1}+y^{-2}+y^{-3}+\cdots
```

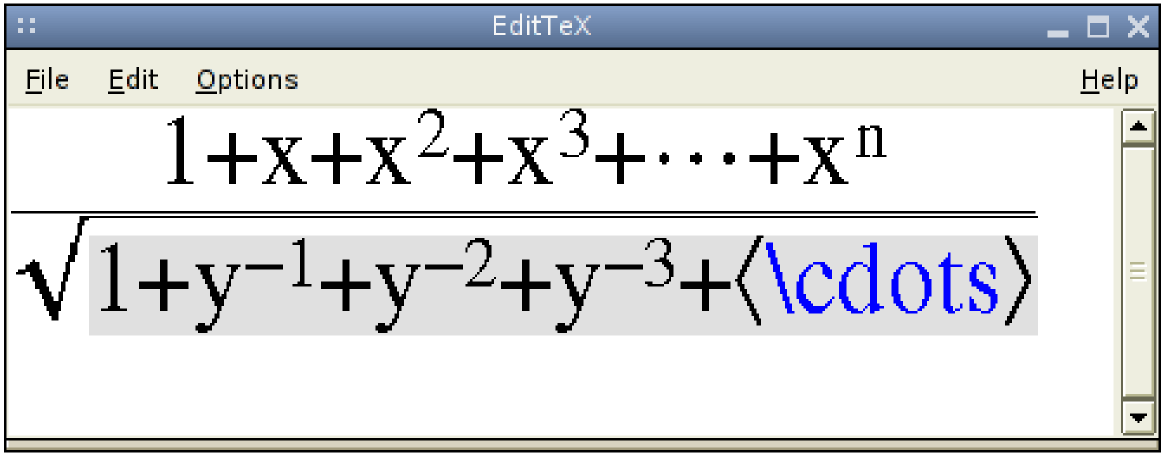


$$\frac{1+x+x^{2}+x^{3}+\cdots+x^{n}}{\sqrt{1+y^{-1}+y^{-2}+y^{-3}+\langle\backslash\mathrm{cdots}\rangle}}$$

EdiTeX

Syntax-directed editing using TeX syntax:

```
\frac{1+x+x^2+x^3+\cdots+x^n}
{\sqrt{1+y^{-1}+y^{-2}+y^{-3}+\cdots
```



$$\frac{1+x+x^{2}+x^{3}+\cdots+x^{n}}{\sqrt{1+y^{-1}+y^{-2}+y^{-3}+\langle\backslash\mathrm{cdots}\rangle}}$$

Concrete syntax

⇒ abstract syntax (via incremental parser)

⇒ MathML (via XSLT).

Embedding

Embedding

Issues:

- there are barriers between different document models;

Embedding

Issues:

- there are barriers between different document models;
- smooth cross-model behaviors (selections, cut & paste) are difficult to implement;

Embedding

Issues:

- there are barriers between different document models;
- smooth cross-model behaviors (selections, cut & paste) are difficult to implement;
- there is no standardized editing behavior for mathematics;

Embedding

Issues:

- there are barriers between different document models;
- smooth cross-model behaviors (selections, cut & paste) are difficult to implement;
- there is no standardized editing behavior for mathematics;
- fine-grained integration in required for decent results (baseline in browsers)

Embedding

Solutions available so far. . .

- `GtkTextView` + U+FFFC trick;
- Bonobo component;
- BoxML.

Embedding

Solutions available so far...

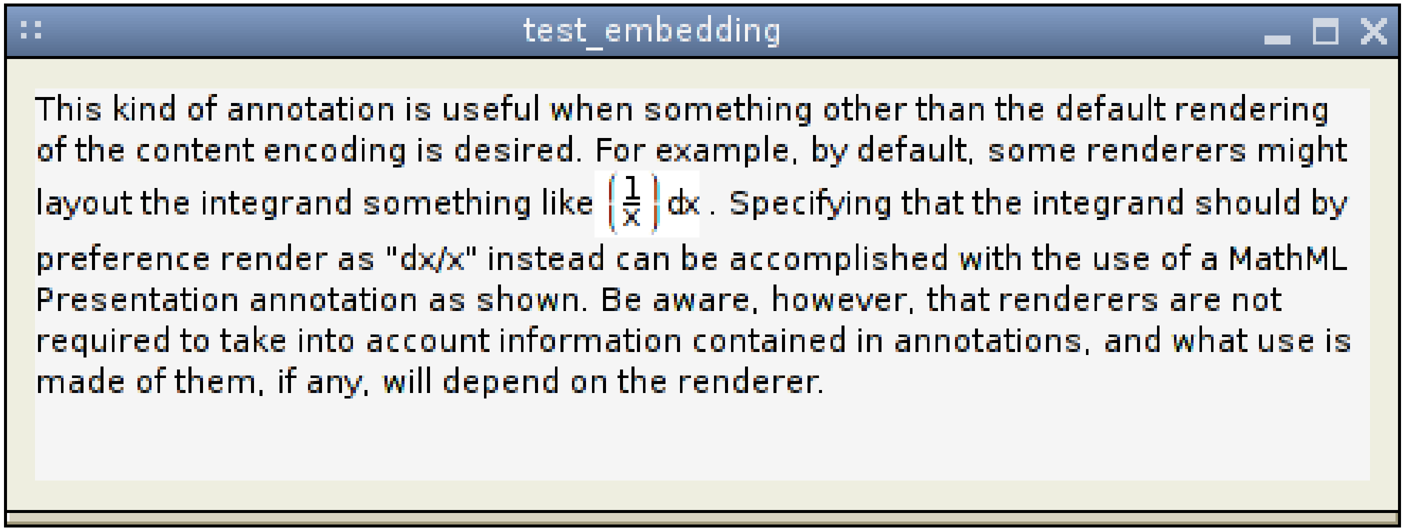
- `GtkTextView` + U+FFFC trick;
- Bonobo component;
- BoxML.

...and problems:

- overhead;
- not precise (missing baseline information);
- ad-hoc.

Embedding in GtkTextView

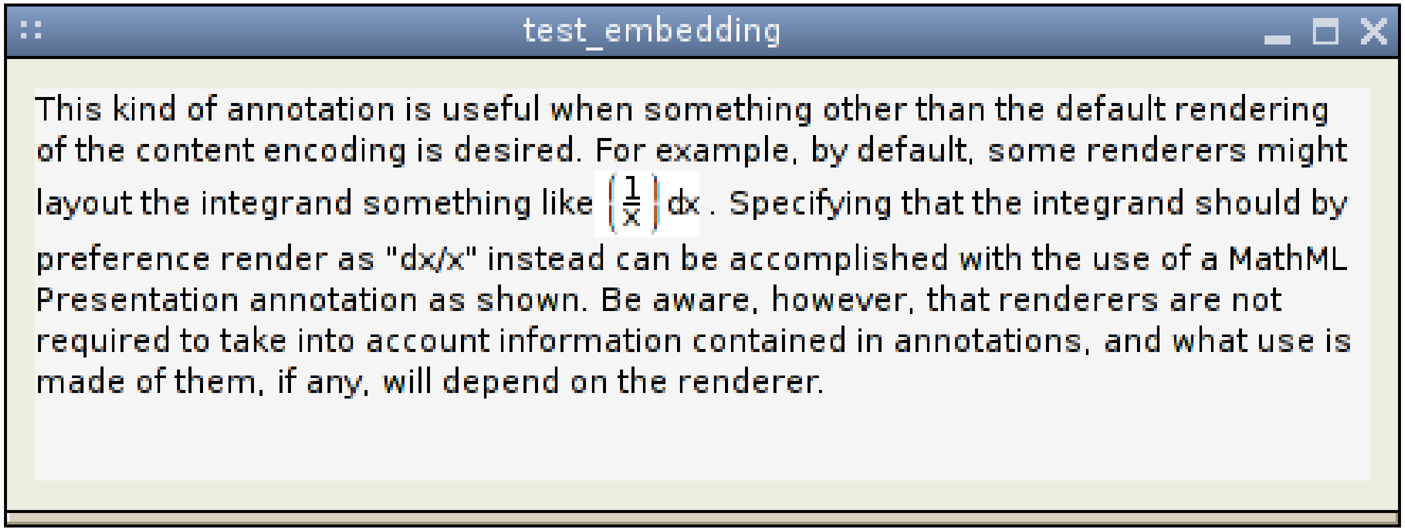
Can be embedded as an anchored child widget within instances of `GtkTextView` widgets:



This kind of annotation is useful when something other than the default rendering of the content encoding is desired. For example, by default, some renderers might layout the integrand something like $\left(\frac{1}{x}\right)dx$. Specifying that the integrand should by preference render as "dx/x" instead can be accomplished with the use of a MathML Presentation annotation as shown. Be aware, however, that renderers are not required to take into account information contained in annotations, and what use is made of them, if any, will depend on the renderer.

Embedding in GtkTextView

Can be embedded as an anchored child widget within instances of `GtkTextView` widgets:



This kind of annotation is useful when something other than the default rendering of the content encoding is desired. For example, by default, some renderers might layout the integrand something like $\left(\frac{1}{x}\right)$dx . Specifying that the integrand should by preference render as "dx/x" instead can be accomplished with the use of a MathML Presentation annotation as shown. Be aware, however, that renderers are not required to take into account information contained in annotations, and what use is made of them, if any, will depend on the renderer.

. . . something like $\left(\frac{1}{x}\right) dx$.

Embedding in GtkTextView

```
buffer = gtk_text_buffer_new(NULL);
gtk_text_buffer_set_text(GTK_TEXT_BUFFER(buffer),
                         "hola  hola", -1);
gtk_text_buffer_get_iter_at_offset(buffer, &start, 5);
anchor = gtk_text_buffer_create_child_anchor(buffer, &start);
t_view = gtk_text_view_new_with_buffer(buffer);
m_view = gtk_math_view_new(NULL, NULL);
gtk_math_view_load_uri(GTK_MATH_VIEW(m_view), "mini.xml");
gtk_math_view_get_bounding_box(GTK_MATH_VIEW(m_view), &m_box);
gtk_text_buffer_get_iter_at_offset(buffer, &start, 5);
gtk_text_buffer_get_iter_at_offset(buffer, &end, 7);
rise_tag = gtk_text_buffer_create_tag(buffer, NULL, "rise",
                     -PANGO_SCALE * m_box.depth, NULL);
gtk_text_buffer_apply_tag(buffer, rise_tag, &start, &end);
gtk_text_view_add_child_at_anchor(GTK_TEXT_VIEW(t_view),
                                  m_view, anchor);
```

Embedding in GtkTextView

```
buffer = gtk_text_buffer_new(NULL);
gtk_text_buffer_set_text(GTK_TEXT_BUFFER(buffer),
                         "hola  hola", -1);
gtk_text_buffer_get_iter_at_offset(buffer, &start, 5);
anchor = gtk_text_buffer_create_child_anchor(buffer, &start);
t_view = gtk_text_view_new_with_buffer(buffer);
m_view = gtk_math_view_new(NULL, NULL);
gtk_math_view_load_uri(GTK_MATH_VIEW(m_view), "mini.xml");
gtk_math_view_get_bounding_box(GTK_MATH_VIEW(m_view), &m_box);
gtk_text_buffer_get_iter_at_offset(buffer, &start, 5);
gtk_text_buffer_get_iter_at_offset(buffer, &end, 7);
rise_tag = gtk_text_buffer_create_tag(buffer, NULL, "rise",
                     -PANGO_SCALE * m_box.depth, NULL);
gtk_text_buffer_apply_tag(buffer, rise_tag, &start, &end);
gtk_text_view_add_child_at_anchor(GTK_TEXT_VIEW(t_view),
                                  m_view, anchor);
```

Embedding in GtkTextView

```
buffer = gtk_text_buffer_new(NULL);
gtk_text_buffer_set_text(GTK_TEXT_BUFFER(buffer),
                         "hola  hola", -1);
gtk_text_buffer_get_iter_at_offset(buffer, &start, 5);
anchor = gtk_text_buffer_create_child_anchor(buffer, &start);
t_view = gtk_text_view_new_with_buffer(buffer);
m_view = gtk_math_view_new(NULL, NULL);
gtk_math_view_load_uri(GTK_MATH_VIEW(m_view), "mini.xml");
gtk_math_view_get_bounding_box(GTK_MATH_VIEW(m_view), &m_box);
gtk_text_buffer_get_iter_at_offset(buffer, &start, 5);
gtk_text_buffer_get_iter_at_offset(buffer, &end, 7);
rise_tag = gtk_text_buffer_create_tag(buffer, NULL, "rise",
                     -PANGO_SCALE * m_box.depth, NULL);
gtk_text_buffer_apply_tag(buffer, rise_tag, &start, &end);
gtk_text_view_add_child_at_anchor(GTK_TEXT_VIEW(t_view),
                                  m_view, anchor);
```

Embedding in GtkTextView

```
buffer = gtk_text_buffer_new(NULL);
gtk_text_buffer_set_text(GTK_TEXT_BUFFER(buffer),
                         "hola  hola", -1);
gtk_text_buffer_get_iter_at_offset(buffer, &start, 5);
anchor = gtk_text_buffer_create_child_anchor(buffer, &start);
t_view = gtk_text_view_new_with_buffer(buffer);
m_view = gtk_math_view_new(NULL, NULL);
gtk_math_view_load_uri(GTK_MATH_VIEW(m_view), "mini.xml");
gtk_math_view_get_bounding_box(GTK_MATH_VIEW(m_view), &m_box);
gtk_text_buffer_get_iter_at_offset(buffer, &start, 5);
gtk_text_buffer_get_iter_at_offset(buffer, &end, 7);
rise_tag = gtk_text_buffer_create_tag(buffer, NULL, "rise",
                     -PANGO_SCALE * m_box.depth, NULL);
gtk_text_buffer_apply_tag(buffer, rise_tag, &start, &end);
gtk_text_view_add_child_at_anchor(GTK_TEXT_VIEW(t_view),
                                  m_view, anchor);
```

The Bonobo Component

Standard Bonobo interfaces implemented:

- `PersistStream`
- `PersistFile`
- `Control`

The Bonobo Component

Standard Bonobo interfaces implemented:

- `PersistStream`
- `PersistFile`
- `Control`

Specific Bonobo interfaces implemented:

- `GtkMathView`

The Bonobo Component

Standard Bonobo interfaces implemented:

- `PersistStream`
- `PersistFile`
- `Control`

Specific Bonobo interfaces implemented:

- `GtkMathView`

Main issue: communication.

Inside Mozilla/Galeon/Epiphany

`mozilla-bonobo` allows Bonobo components with MIME-type to be used as plugins.

Inside Mozilla/Galeon/Epiphany

`mozilla-bonobo` allows Bonobo components with MIME-type to be used as plugins.

`GtkMathView` is faster than the MathML rendering engine in Gecko.

Inside Mozilla/Galeon/Epiphany

`mozilla-bonobo` allows Bonobo components with MIME-type to be used as plugins.

`GtkMathView` is faster than the MathML rendering engine in Gecko.

- + possible to execute JavaScript code;
- + possible to open new pages;
- – no baseline alignment;
- – no size negotiation;
- – border around the plugin.

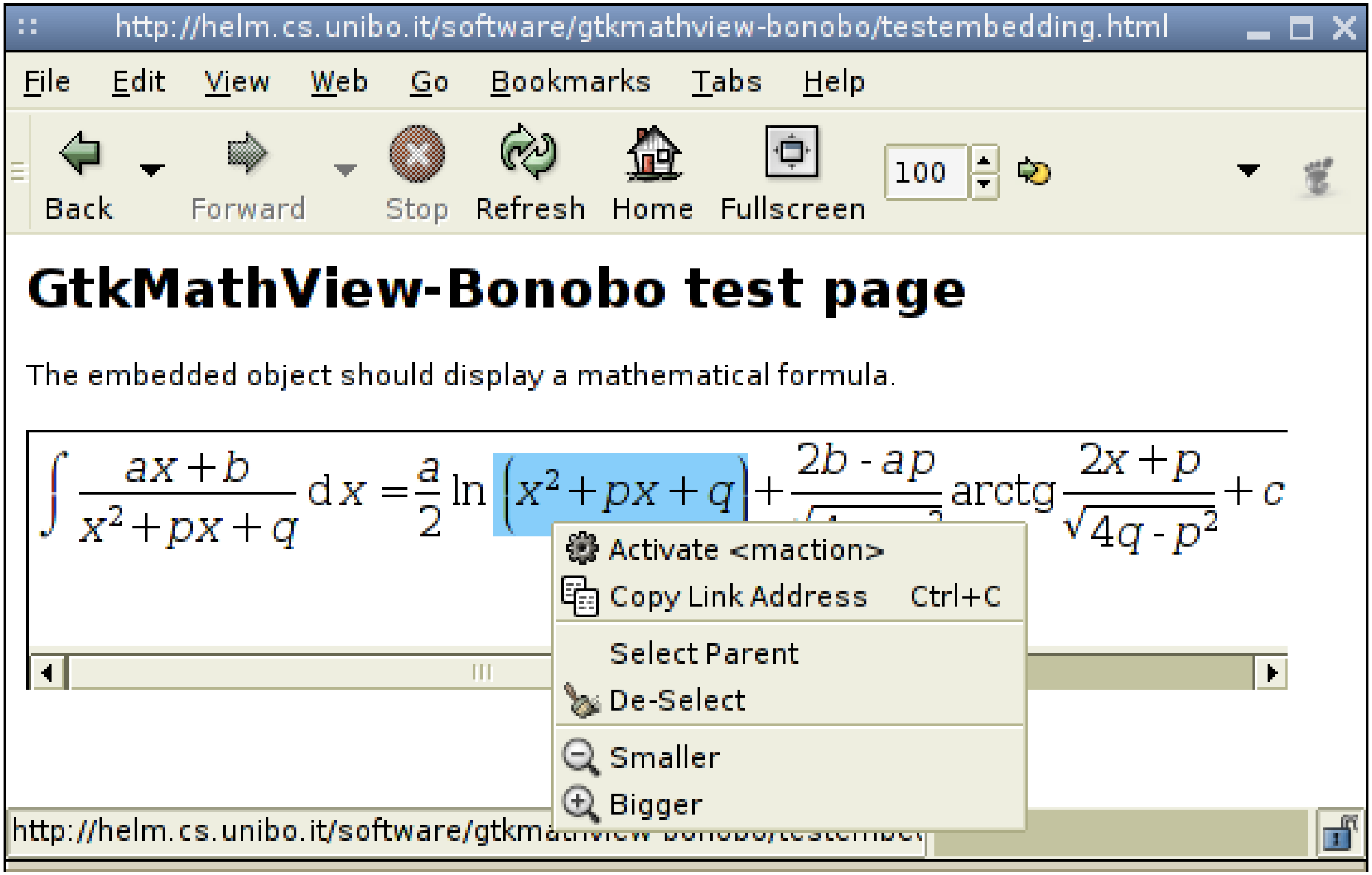
Inside Mozilla/Galeon/Epiphany

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <body>
    <h1>GtkMathView-Bonobo test page</h1>
    <p>The embedded object should display a mathematical formula.
    <object name="MATH" data="#math1" width="500" height="100"
            type="application/mathml+xml">
      <math id="math1" display="block"
            xmlns="http://www.w3.org/1998/Math/MathML">
        <mfrac> <mi>x</mi> <mn>2</mn> </mfrac>
        <mo>=</mo>
        <mtext>?</mtext>
      </math>
    </object>
  </body>
</html>
```

Inside Mozilla/Galeon/Epiphany

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <body>
    <h1>GtkMathView-Bonobo test page</h1>
    <p>The embedded object should display a mathematical formula.
    <object name="MATH" data="#math1" width="500" height="100"
            type="application/mathml+xml">
      <math id="math1" display="block"
            xmlns="http://www.w3.org/1998/Math/MathML">
        <mfrac> <mi>x</mi> <mn>2</mn> </mfrac>
        <mo>=</mo>
        <mtext>?</mtext>
      </math>
    </object>
  </body>
</html>
```

Inside Mozilla/Galeon/Epiphany



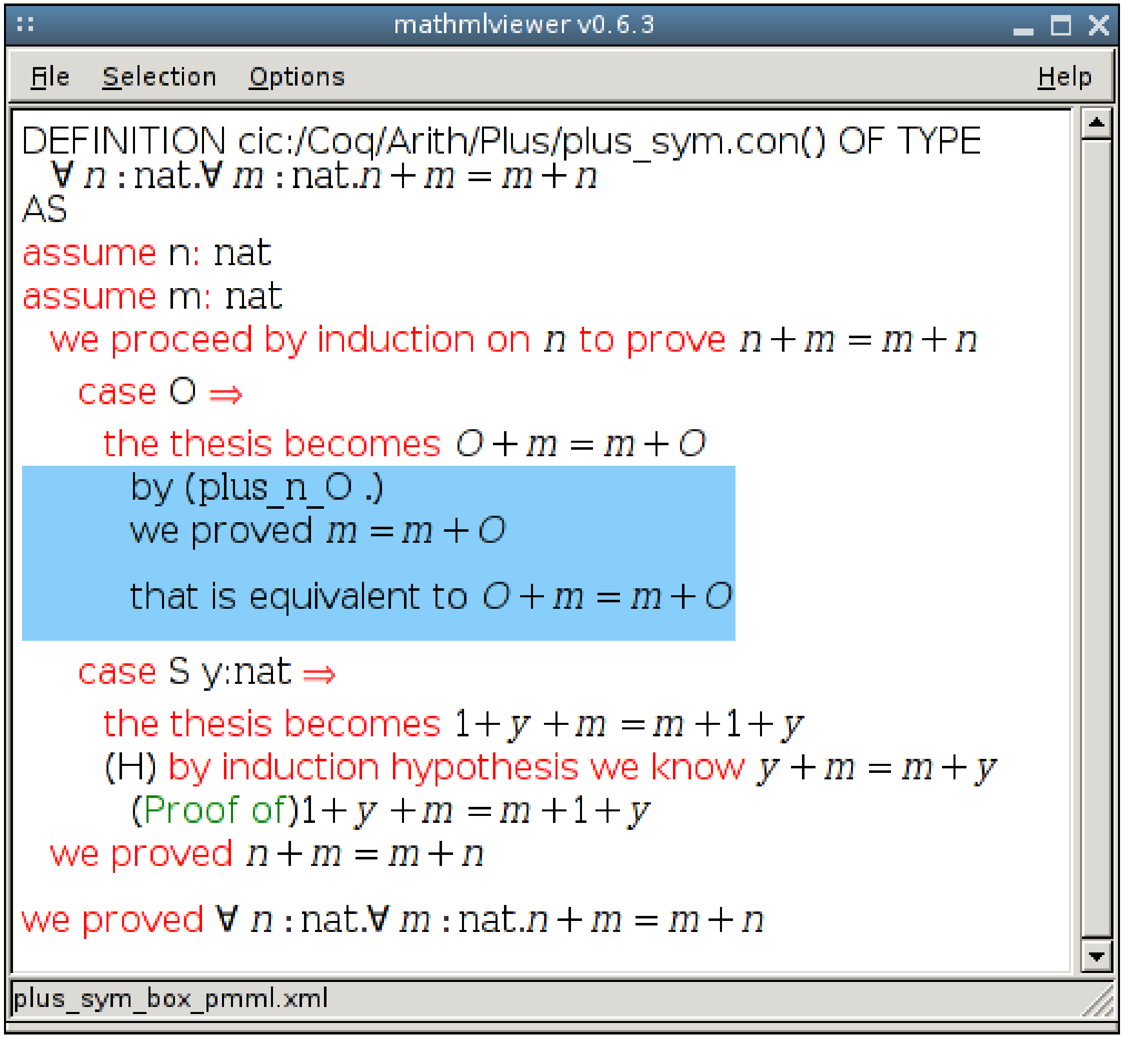
GtkMathView-Bonobo test page

The embedded object should display a mathematical formula.

$$\int \frac{ax+b}{x^2+px+q}\,\mathrm{d}x = \frac{a}{2}\ln\left|x^2+px+q\right| + \frac{2b-ap}{\sqrt{4q-p^2}}\,\mathrm{arctg}\,\frac{2x+p}{\sqrt{4q-p^2}} + c$$

Embedding in BoxML

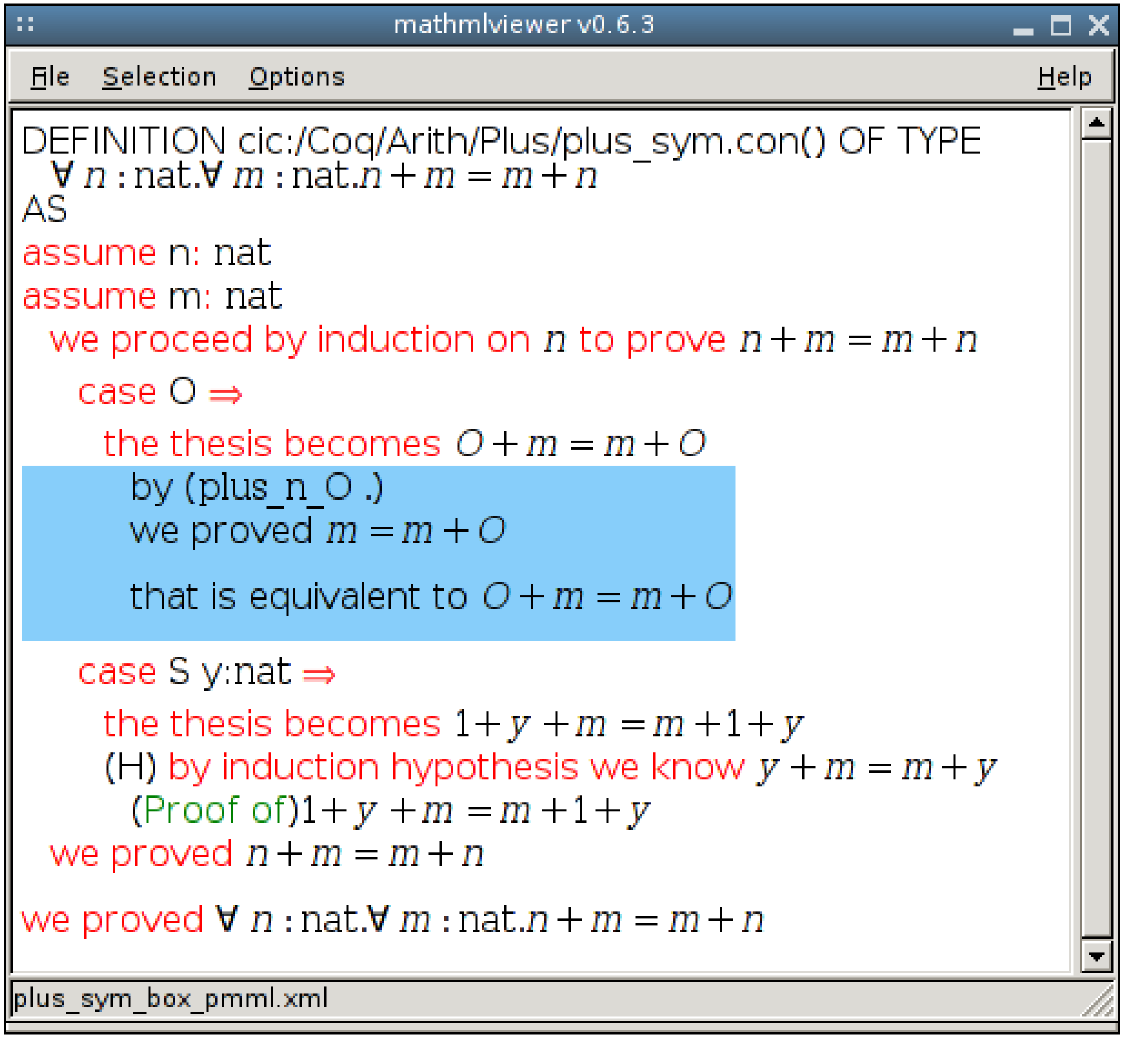
Embedding in BoxML



DEFINITION cic:/Coq/Arith/Plus/plus_sym.con() OF TYPE
$\forall n:\mathrm{nat}.\forall m:\mathrm{nat}.n+m=m+n$
AS
assume n: nat
assume m: nat
we proceed by induction on n to prove $n+m=m+n$
case O ⇒
the thesis becomes $O+m=m+O$
by (plus_n_O .)
we proved $m=m+O$
that is equivalent to $O+m=m+O$
case S y:nat ⇒
the thesis becomes $1+y+m=m+1+y$
(H) by induction hypothesis we know $y+m=m+y$
(Proof of)$1+y+m=m+1+y$
we proved $n+m=m+n$
we proved $\forall n:\mathrm{nat}.\forall m:\mathrm{nat}.n+m=m+n$

plus_sym_box_pmml.xml

Embedding in BoxML



DEFINITION cic:/Coq/Arith/Plus/plus_sym.con() OF TYPE
  $\forall n:\text{nat}.\forall m:\text{nat}.n+m=m+n$
AS
assume n: nat
assume m: nat
  we proceed by induction on n to prove $n+m=m+n$
    case O ⇒
      the thesis becomes $O+m=m+O$
        by (plus_n_O .)
        we proved $m=m+O$
        that is equivalent to $O+m=m+O$
    case S y:nat ⇒
      the thesis becomes $1+y+m=m+1+y$
      (H) by induction hypothesis we know $y+m=m+y$
        (Proof of)$1+y+m=m+1+y$
  we proved $n+m=m+n$
we proved $\forall n:\text{nat}.\forall m:\text{nat}.n+m=m+n$

Inspired by $\TeX$ boxes and languages for pretty-printing.

Embedding in BoxML

```
<b:box xmlns:b="http://helm.cs.unibo.it/2003/BoxML">
 <b:v xmlns:m="http://www.w3.org/1998/Math/MathML">
  <b:v>
   <b:h>
    <b:text>DEFINITION cic:/Coq/Arith/Plus/plus_sym.con(</b:t
    <b:text>) OF TYPE</b:text>
   </b:h>
   <b:h>
    <b:space width="1em"/>
    <b:h>
     <b:obj>
      <m:semantics>
       <m:mrow>
        <m:mrow>
         <m:mo mathcolor="blue">&#x2200;</m:mo>
         <m:mi>n</m:mi>
```

Embedding in BoxML

```
<b:box xmlns:b="http://helm.cs.unibo.it/2003/BoxML">
  <b:v xmlns:m="http://www.w3.org/1998/Math/MathML">
    <b:v>
      <b:h>
        <b:text>DEFINITION cic:/Coq/Arith/Plus/plus_sym.con(</b:t
        <b:text>) OF TYPE</b:text>
      </b:h>
      <b:h>
        <b:space width="1em"/>
        <b:h>
          <b:obj>
            <m:semantics>
              <m:mrow>
                <m:mrow>
                  <m:mo mathcolor="blue">&#x2200;</m:mo>
                  <m:mi>n</m:mi>
```

BoxML Summary

Element	Attributes	Description
box		root element
action	actiontype, selection	alternative renderings
at	x, y	element at fixed coordinates
h		horizontal box
ink	color, width, height, depth	solid box
layout	width, height, depth	fixed layout box
space	width, height, depth	empty box
v	enter, exit	vertical box
text	color, background, size	text
obj		embedded MathML

Internals

Architecture

Very much compiler-like:

- parse the source document model;
- create internal representation (AST);
- translate the internal representation into low-level representation (area model);
- render the area tree on the output medium.

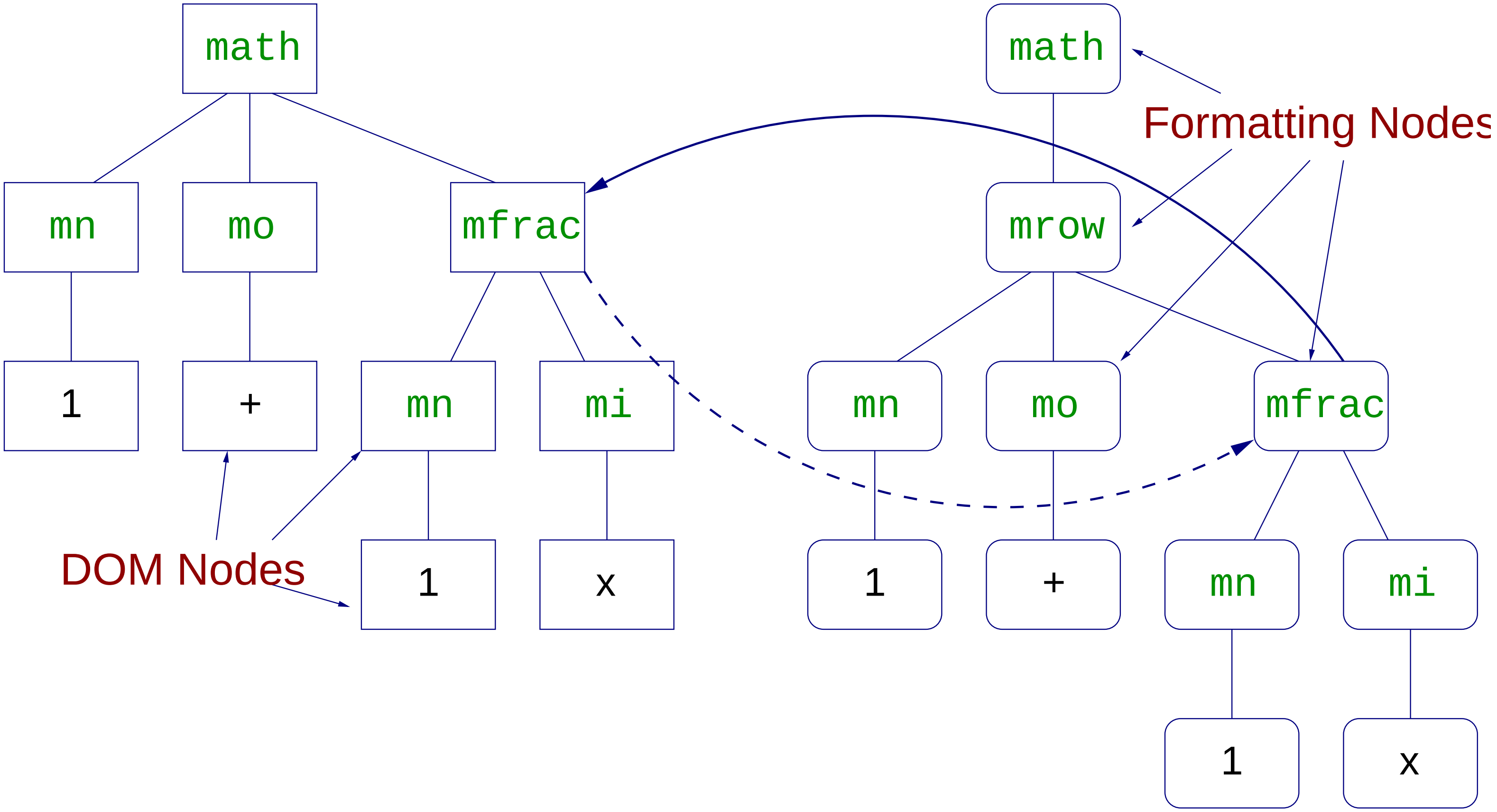
Architecture

Very much compiler-like:

- parse the source document model;
- create internal representation (AST);
- translate the internal representation into low-level representation (area model);
- render the area tree on the output medium.

. . . but context-sensitive “code generation” phase.

Architecture



Source MathML tree

Formatting tree

Feeding the widget

In principle: use the GNOME DOM engine

`Gdome2` `http://gdome2.cs.unibo.it`, for C

`GMetaDOM` `http://gmetadom.sourceforge.net`, for C++

Feeding the widget

In principle: use the GNOME DOM engine

`Gdome2` `http://gdome2.cs.unibo.it`, for C

`GMetaDOM` `http://gmetadom.sourceforge.net`, for C++

In practice: there is disagreement as to how to deliver XML content to the widgets:

- plain text;
- `libxml2` API;
- private DOM.

Using the GNOME DOM engine

Drawbacks:

- hides `libxml2` functionalities that people want;

Using the GNOME DOM engine

Drawbacks:

- hides `libxml2` functionalities that people want;
- increases dependencies (two more libraries);

Using the GNOME DOM engine

Drawbacks:

- hides `libxml2` functionalities that people want;
- increases dependencies (two more libraries);
- increases both memory occupation and processor overhead;
- forces the document tree to be loaded entirely.

Using the GNOME DOM engine

Drawbacks:

- hides `libxml2` functionalities that people want;
- increases dependencies (two more libraries);
- increases both memory occupation and processor overhead;
- forces the document tree to be loaded entirely.

Using `libxml2`?

A frontend for everybody

Solution: use C++ templates for the model navigation strategy, the actual document model is a template parameter.

A frontend for everybody

Solution: use C++ templates for the model navigation strategy, the actual document model is a template parameter.

- + performance (no virtual methods)
- + adaptability
- + there is no abstract interface to implement
- – slow compilation
- – cannot change frontend at runtime

A frontend for everybody

Solution: use C++ templates for the model navigation strategy, the actual document model is a template parameter.

- + performance (no virtual methods)
- + adaptability
- + there is no abstract interface to implement
- – slow compilation
- – cannot change frontend at runtime

MSXML frontend is now trivial to add.

Porting `GtkMathView`

How to port the widget to a different font family?

Porting `GtkMathView`

How to port the widget to a different font family?

- implement a graphic device;

Porting `GtkMathView`

How to port the widget to a different font family?

- implement a graphic device;
- implement shapers;

Porting `GtkMathView`

How to port the widget to a different font family?

- implement a graphic device;
- implement shapers;
- implement platform-specific areas.

Porting `GtkMathView`

How to port the widget to a different font family?

- implement a graphic device;
- implement shapers;
- implement platform-specific areas.

Fonts for math have many (implicit) parameters with no standardized names. See Knuth's CM fonts.

Must take advantage of these parameters without compromising modularity: the separation of the backend from the engine is delicate.

Areas

Area base class and main (pure) virtual methods:

```
class Area : public Object {
  virtual void render(class RenderingContext&,
                      const scaled& x, const scaled& y) const;
  virtual BoundingBox box(void) const;
  virtual scaled leftEdge(void) const;
  virtual scaled rightEdge(void) const;
  virtual AreaIndex size(void) const;
  virtual AreaRef node(AreaIndex) const;
  virtual void origin(AreaIndex, class Point&) const;
  virtual bool indexOfPosition(const scaled&, const scaled&,
                               CharIndex&) const;
  virtual bool positionOfIndex(CharIndex, class Point*,
                               BoundingBox*) const;
};
```

Areas

Area base class and main (pure) virtual methods:

```
class Area : public Object {
  virtual void render(class RenderingContext&,
                      const scaled& x, const scaled& y) const;
  virtual BoundingBox box(void) const;
  virtual scaled leftEdge(void) const;
  virtual scaled rightEdge(void) const;
  virtual AreaIndex size(void) const;
  virtual AreaRef node(AreaIndex) const;
  virtual void origin(AreaIndex, class Point&) const;
  virtual bool indexOfPosition(const scaled&, const scaled&,
                               CharIndex&) const;
  virtual bool positionOfIndex(CharIndex, class Point*,
                               BoundingBox*) const;
};
```

Areas

Area base class and main (pure) virtual methods:

```
class Area : public Object {
  virtual void render(class RenderingContext&,
                      const scaled& x, const scaled& y) const;
  virtual BoundingBox box(void) const;
  virtual scaled leftEdge(void) const;
  virtual scaled rightEdge(void) const;
  virtual AreaIndex size(void) const;
  virtual AreaRef node(AreaIndex) const;
  virtual void origin(AreaIndex, class Point&) const;
  virtual bool indexOfPosition(const scaled&, const scaled&,
                               CharIndex&) const;
  virtual bool positionOfIndex(CharIndex, class Point*,
                               BoundingBox*) const;
};
```

Areas

Area base class and main (pure) virtual methods:

```
class Area : public Object {
  virtual void render(class RenderingContext&,
                      const scaled& x, const scaled& y) const;
  virtual BoundingBox box(void) const;
  virtual scaled leftEdge(void) const;
  virtual scaled rightEdge(void) const;
  virtual AreaIndex size(void) const;
  virtual AreaRef node(AreaIndex) const;
  virtual void origin(AreaIndex, class Point&) const;
  virtual bool indexOfPosition(const scaled&, const scaled&,
                               CharIndex&) const;
  virtual bool positionOfIndex(CharIndex, class Point*,
                               BoundingBox*) const;
};
```

Areas

Area base class and main (pure) virtual methods:

```
class Area : public Object {
  virtual void render(class RenderingContext&,
                      const scaled& x, const scaled& y) const;
  virtual BoundingBox box(void) const;
  virtual scaled leftEdge(void) const;
  virtual scaled rightEdge(void) const;
  virtual AreaIndex size(void) const;
  virtual AreaRef node(AreaIndex) const;
  virtual void origin(AreaIndex, class Point&) const;
  virtual bool indexOfPosition(const scaled&, const scaled&,
                               CharIndex&) const;
  virtual bool positionOfIndex(CharIndex, class Point*,
                               BoundingBox*) const;
};
```

Pango and math (mango?)

Pango and math (mango?)

- Sometimes we need to combine characters in math-sensible ways ($\widehat{\longrightarrow}$).

Pango and math (mango?)

- Sometimes we need to combine characters in math-sensible ways ($\widehat{\longrightarrow}$).
- Pango does not always support well combining characters occurring frequently in math ($\neq$).

Pango and math (mango?)

- Sometimes we need to combine characters in math-sensible ways ($\widehat{\longrightarrow}$).

- Pango does not always support well combining characters occurring frequently in math ($\neq$).

- Stretchable characters rendered with compound glyphs: need to compute `PangoGlyphStrings` manually.

Pango and math (mango?)

- Sometimes we need to combine characters in math-sensible ways ($\widehat{\longrightarrow}$).

- Pango does not always support well combining characters occurring frequently in math ($\neq$).

- Stretchable characters rendered with compound glyphs: need to compute `PangoGlyphStrings` manually.

- some glyphs are not available as Unicode characters ($\underbrace{1+x+x^2}$).

Conclusion

Version history

	GTK	editing	PostScript	tables	alignment	frontend
0.3.1	1.2	no	yes	yes	yes	`MiniDOM/GMetaDOM`
0.4.3	1.2	yes	yes	yes	?	`GMetaDOM`
0.5.3	2.2	yes	yes	yes	no	`GMetaDOM`
0.6.3	2.4	yes	no	no	no	any

Also available: OCaml bindings up to version 0.5.2.

What is missing?

- Complete MathML support (tables, alignment);

What is missing?

- Complete MathML support (tables, alignment);
- re-implement PS/PDF exportation;

What is missing?

- Complete MathML support (tables, alignment);
- re-implement PS/PDF exportation;
- state-of-the-art editor (WYSIWYG + ideas from EdiTEX);

What is missing?

- Complete MathML support (tables, alignment);
- re-implement PS/PDF exportation;
- state-of-the-art editor (WYSIWYG + ideas from EdiTEX);
- integration with widgets for HTML;

What is missing?

- Complete MathML support (tables, alignment);
- re-implement PS/PDF exportation;
- state-of-the-art editor (WYSIWYG + ideas from EdiTEX);
- integration with widgets for HTML;
- integration with AbiWord;

What is missing?

- Complete MathML support (tables, alignment);
- re-implement PS/PDF exportation;
- state-of-the-art editor (WYSIWYG + ideas from EdiTEX);
- integration with widgets for HTML;
- integration with AbiWord;
- volunteers.

What is missing?

- Complete MathML support (tables, alignment);
- re-implement PS/PDF exportation;
- state-of-the-art editor (WYSIWYG + ideas from EdiTEX);
- integration with widgets for HTML;
- integration with AbiWord;
- volunteers.

Thank you