Publishing on the Web, today and tomorrow

Michel Goossens, CERN

JACoW99

- current problems with HTML;
- HTML 4, a step forward;
- the extensible markup language (XML) effort in general;
- a closer look at the XML and XSL syntax;
- math on the Web: images and Java today, MathML tomorrow;
- conslusion.

CERN

Why HTML has problems

- *Invalid* HTML*:* applications produce invalid HTML or introduce vendor-specific extensions; browsers do not reject invalid HTML.
- *Broken links:* problem of deleted or moved Web pages (use URL).
- *Fixed grammar:* HTML is a *fixed* SGML DTD (language syntax); no *standard* way to extend, adapt, renew the language.
- *Limited support for meta-data:* what about keywords, search-engines, re-use of the same source file.
- *Absence of structural tags:* HTML tags control mainly appearance not structure; navigation difficult.
- *Data exchange difficulties:* HTML aimed at presentation (Latin1 codes); difficult for data mining, internationalisation, exchange.
- *Absence of modern features:* Web changes rapidly; HTML must adapt quickly to new technology; XHTML and modules.

CERN

HTML, the way ahaead

- *Extend* HTML*:* HTML 4 (4.01) recommendations (Dec. 1997, Nov. 1999), HTML 4. is end of the road (May 1998).
- *Extensible Markup Language (*XML*):* XML and its tools offer better application-specificity and data organisation.
- XHTML*:* Modular XML application of HTML: version 1.1 (April 2000), and version 2.0, replacing Images and Anchors with XLink (July 2000). XHTMLBasic for thin clients (April 2000).
- *Cascading Style Sheets:* CSS1 and CSS2 recommendations: decouple presentation and content.
- *Document Object Model (*DTD*):* programmatic access to HTML (XML) elements as object data, exposing properties and methods.

Problem: Almost no browsers support XML directly at present.

CERN

Extensible Markup Languages

- Since mid 1996 W3C SGML (now renamed XML) Working Group had been developing a “system for defining, validating, and sharing document formats on the Web”.
- XML 1.0 became a W3C Recommendation on 10 February 1998.
- It is the first in a series of standards.
- Others are XSL, XPointer, Xlink, MathML.
- Much other work is going on:
  Chemical Markup Language (CML), Object Data Model (DOM), Music Markup Language (MusicML), Scalable Vector Graphics (SVG), Resource Description Framework (RDF), Weather Observation Markup Format (OMF), Many Ecommerce initiatives, etc.

CERN

What is XML?

- light-weight subset of SGML (ISO/IEC 8879:1986);
- meta-language to described logical structure of document;
- well integrated with the Internet;
- internationalisation built in from the start since basic languages is Unicode (UTF-16 and UTF-8);
- easy to learn, modify and implement;
- logical relations between elements described in *Document Type Definition* (DTD);
- documents can be *correct* (checked w.r.t. DTD);
- documents can be *well-formed* (no DTD needed, nesting correct).

CERN

Structure of an XML document

- simplest complete document (*well-formed*);

```
<coolxml>XML is a cool idea!</coolxml>
```

- the same document in its *valid* form:

```
<?xml version="1.0" standalone="yes"?> <!-- XML PI -->
<!DOCTYPE coolxml [
  <!ELEMENT coolxml (#PCDATA)>
]>
<coolxml>XML is a cool idea!</coolxml>
```

It has three parts:

1. XML processing instruction (version, encoding, stand-alone);
2. document type declaration (internal and external subsets);
3. document instance.

CERN

The Document Type Definition (DTD)

- defines grammar of our *little* language;
- uses its own syntax (*XML-data*, Xschema, DCD propose XML format);
- *elements*: visual structural markup components;

```
<!ELEMENT pome      (prambule, corps)>
<!ELEMENT prambule  (titre, recueil?, auteur, date?)>
<!ELEMENT titre      (#PCDATA)>
```

- *attributes*: properties of elements;

```
<!ATTLIST line  newpara (yes|no)  "no"
                indent  CDATA     "0mm"
                text    CDATA     #REQUIRED
                color   NMTOKEN   #FIXED "red">
```

- *entities*: internal and external foreign material.

CERN

Internal entities

- defined in DTD, have no associated storage object, always parsed;
- abbreviations (*general entity*).

```
<!ENTITY MML "Mathematical Markup Language">
```

- special characters, accents or symbols (*general character entity*).

```
<!ENTITY gt    CDATA "&#62;">
<!ENTITY cyrya CDATA "&#x044f;">
```

  Predefined entities: `lt` (<), `gt` (>), `amp` (&), `apos` (’), and `quot` (").

- definition of variables in DTD (*parameter entity*).

```
<!ENTITY % list "UL | OL |  DIR | MENU">
```

- definition must precede reference: `&symb;` or `%list;`;
- references can be used inside definitions:

```
<!ENTITY XMLS "&MML; and other extensible languages">
```

CERN

External entities

- those that are not *internal*;
- reference data external to current document;
- data from external document can be parsed or declared `NDATA` (unparsed, e.g., bitmap image or binary file);
- include external file with `"SYSTEM"` keyword and followed by URL (*Universal Resource Identifier*).

  ```
  <!ENTITY article SYSTEM "/usr/goossens/articles/xmlart.xml">
  ```

  Contents of XML source file included (after parsing) with `&article;`
- include external file with `"PUBLIC"` keyword and followed by public identifier literal, and system literal (URL):

  ```
  <!ENTITY % html4-strict PUBLIC "-//W3C//DTD HTML 4.0//EN"
           "http://www.w3.org/TR/REC-html40/strict.dtd">
  ```

CERN

The XML application first tries to build URL using public name `-//W3C//DTD HTML 4.0//EN`, e.g., via the `catalog` file proposed by OASIS (Organization for the Advancement of Structured Information Standards), otherwise uses explicit URL at end.

- non-parsable data (GIF or JPEG images, binary files) need the definition of a notation to handle data type

```
<!NOTATION GIF SYSTEM
   "c:\Program Files\Internet Explorer\Ie4.dll" >
```

A gif image is then declared with:

```
<!ENTITY xmlfig1 SYSTEM
  "http://www.myserver.edu/book-files/figures/xmlfig1"
  NDATA GIF >
```

and later included, for instance, with:

```
<IMG url="&xmlfig1;">
```

CERN

Doing it with style: XSL

- Historically two approaches : CSS ("HTML-world") *versus* DSSSL ("computer professionals");
- CSS: from simple to more complex (CSS1, CSS2), mainly for HTML-based web applications;
- DSSSL-o: subset of DSSSL, allows for transformations between document types, can generate toc's, indexes, page headers, . . .
- XSL (*Extensible Style Language*) WG set up on January 23rd 1998.
- Uses its own language (not XML), so that is can be used inside attributes.
- Contains XPath (addressing points in XML documents, common with XPointer), XSLT (transformation), and XLSF (formatting objects).
- XPath and XSLT became W3C recommendations on 16 November 1999.

CERN

A small example file

A simple source file

```
<!DOCTYPE invitation SYSTEM "invitation.dtd">
<invitation>
<front>
<to>Anna, Bernard, Didier, Johanna</to>
<date>Next Friday Evening at 8 pm</date>
<where>The Web Cafe</where>
<why>My first XML baby</why>
</front>
<body>
<par>I would like to invite you all to celebrate
the birth of <emph>Invitation</emph>, my
first XML document child.</par>
<par>Please do your best to come and join me next Friday
evening. And, do not forget to bring your friends.</par>
<par>I <emph>really</emph> look forward to see you soon!</par>
</body>
<back>
<signature>Michel</signature>
</back>
</invitation>
```

. . . and its DTD

```
<!-- invitation DTD   -->
<!ELEMENT invitation (front, body, back) >
<!ELEMENT front      (to, date, where, why?) >
<!ELEMENT date       (#PCDATA) >
<!ELEMENT to         (#PCDATA) >
<!ELEMENT where      (#PCDATA) >
<!ELEMENT why        (#PCDATA) >
<!ELEMENT body       (par+) >
<!ELEMENT par        (#PCDATA|emph)* >
<!ELEMENT emph       (#PCDATA) >
<!ELEMENT back       (signature) >
<!ELEMENT signature  (#PCDATA) >
```

CERN

Doing it with XSL

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:fo="http://www.w3.org/1999/XSL/Format"
                default-space="">

<xsl:variable name="PageMarginTop">75pt</xsl:variable>
<xsl:variable name="PageMarginBottom">125pt</xsl:variable>
<xsl:variable name="PageMarginLeft">80pt</xsl:variable>
<xsl:variable name="PageMarginRight">150pt</xsl:variable>
<xsl:variable name="BodySize">12pt</xsl:variable>

<xsl:template match='/'>
  <fo:root xmlns:fo="http://www.w3.org/XSL/Format/1.0">
    <fo:layout-master-set>
      <fo:simple-page-master
        page-master-name="allpages"
        margin-top="{$PageMarginTop}"
        margin-bottom="{$PageMarginBottom}"
        margin-left="{$PageMarginLeft}"
        margin-right="{$PageMarginRight}">
        <fo:region-body margin-bottom="100pt"/>
        <fo:region-after extent="25pt"/>
      </fo:simple-page-master>
      </fo:layout-master-set>
      <fo:page-sequence>
        <fo:sequence-specification>
          <fo:sequence-specifier-repeating
            page-master-first="allpages"
            page-master-repeating="allpages"/>
        </fo:sequence-specification>
        <fo:flow font-family="serif">
          <xsl:apply-templates/>
        </fo:flow>
      </fo:page-sequence>
  </fo:root>
</xsl:template>

<xsl:template match="invitation/front">
<fo:block font-family="sans-serif"   font-size="24pt"
          text-align-last="centered" font-weight="bold"
          space-after.optimum="24pt">
  <xsl:text>INVITATION</xsl:text>
</fo:block>
 ...
<xsl:template match="invitation/back">
<fo:block space-before.optimum="12pt}"
          font-weight="bold" text-align-last="end">
   <xsl:text>From: </xsl:text>
   <xsl:value-of select="signature"/>
</fo:block>
</xsl:template>

</xsl:stylesheet>
```

CERN

Result with FOP and PassiveTEX

$$\text{XML} \xrightarrow{\text{XSL}} \text{FO} \xrightarrow{\text{FOP}} \text{PDF}$$

INVITATION

To: Anna, Bernard, Didier, Johanna
When: Next Friday Evening at 8 pm
Venue: The Web Cafe
Occasion: My first XML baby

I would like to invite you all to celebrate the birth of *Invitation* , my first XML document child.

Please do your best to come and join me next Friday evening. And, do not forget to bring your friends.

I *really* look forward to see you soon!

From: Michel

$$\text{XML} \xrightarrow{\text{XSL}} \text{FO} \xrightarrow{\text{PassiveTEX}} \text{PDF}$$

INVITATION

To: Anna, Bernard, Didier, Johanna
When: Next Friday Evening at 8 pm
Venue: The Web Cafe
Occasion: My first XML baby

I would like to invite you all to celebrate the birth of *Invitation*, my first XML document child.

Please do your best to come and join me next Friday evening. And, do not forget to bring your friends.

I *really* look forward to see you soon!

From: Michel

CERN

MathML: An XML vocabulary for math

MathML will make the Web even better for educational, scientific and technical materials. It also has the potential to make mathematics accessible to those with visual disabilities. It will allow mathematical content to be reused and exchanged with technical computing systems for further manipulation.

W3C Director Tim Berners-Lee

7 April, 1998

(release of MathML as a W3C Recommendation).

MathML contains elements of three categories: presentation, content, and interface elements.

CERN

Aims of MathML

- Encode mathematical material suitable for teaching and scientific communication at all levels;
- encode both mathematical notation and mathematical meaning;
- facilitate conversion to and from other math formats, both presentational and semantic. Output formats should include: graphical displays, speech synthesizers, computer algebra, other math layout languages (TEX), plain text displays (VT100 emulators), print media, braille. Such conversions may entail loss of information.
- allow information intended for specific renderers and applications;
- support efficient browsing for lengthy expressions;
- provide for extensibility;
- be well suited to template and other math editing techniques;
- be human legible, and simple for software to generate and process.

CERN

Presentation and content markup

- **Presentation markup**
  - describe mathematical notation structure;
  - 28 elements, more than 50 attributes;
  - generally, each presentation element corresponds to a single kind of “layout schema” (two-dimensional notational devide, such as row, superscript, underscript);
  - variants are expressed via attributes.
- **Content markup**
  - directly describe mathematical objects;
  - about 75 elements and about a dozen attributes;
  - Tokens, basic content, arithmetic, algebra, logic, relations, calculus, set theory, sequences, series, trigonometry, statistics, linear algebra, semantic mappings.

CERN

A first MathML Example

$$x^2 + 4x + 4 = 0$$

Presentation Markup

```
<mrow><!-- horizontal alignment -->
  <mrow>
    <msup>      <!-- superscript-->
      <mi>x</mi><!-- identifier -->
      <mn>2</mn><!-- number     -->
    </msup>
    <mo>+</mo>  <!-- operator   -->
    <mrow>
      <mn>4</mn>
      <mo>&InvisibleTimes;</mo>
      <mi>x</mi>
    </mrow>
    <mo>+</mo>
    <mn>4</mn>
  </mrow>
  <mo>=</mo>
  <mn>0</mn>
</mrow>
```

Content Markup

```
<reln>
  <eq/>
  <apply>
    <plus/>
    <apply>
      <power/>
      <ci>x</ci>
      <cn>2</cn>
    </apply>
    <apply>
      <times/>
      <cn>4</cn>
      <ci>x</ci>
    </apply>
    <cn>4</cn>
  </apply>
  <cn>0</cn>
</reln>
```

CERN

Another MathML example

$$A = \begin{bmatrix} x & y \\ z & w \end{bmatrix}$$

Presentation Markup

```
<mrow>
  <mi>A</mi>
  <mo>=</mo>
  <mfenced open="[" close="]">
    <mtable><!-- table or matrix -->
      <mtr> <!-- table row       -->
         <mtd><mi>x</mi></mtd><-- table -->
         <mtd><mi>y</mi></mtd><-- entry -->
      </mtr>
      <mtr>
         <mtd><mi>z</mi></mtd>
         <mtd><mi>w</mi></mtd>
      </mtr>
    </mtable>
  </mfenced>
</mrow>
```

Content Markup

```
<reln>
  <eq/>
  <ci>A</ci>
  <matrix>
    <matrixrow>
      <ci>x</ci>
      <ci>y</ci>
    </matrixrow>
    <matrixrow>
      <ci>z</ci>
      <ci>w</ci>
    </matrixrow>
  </matrix>
</reln>
```

CERN

Document strategies for the Web

- Millions of document exist in electronic form. *Reuse* of the large investment in this knowledge base is an important point.
- *New* documents can exploit the advantages of the latest technologies and adapt readily to the target audience (e-commerce, scientific, dictionaries, etc.).
- *Several presentations* of the same source are often a must, and *ad hoc* and different techniques have to be used to optimally exploit the display possibilities of the medium (browser, print, audio-video).
- Their exist a *variety* of application domains. Scientific articles, financial data, or ancient Greek poetry need different tools and target different audience.

CERN

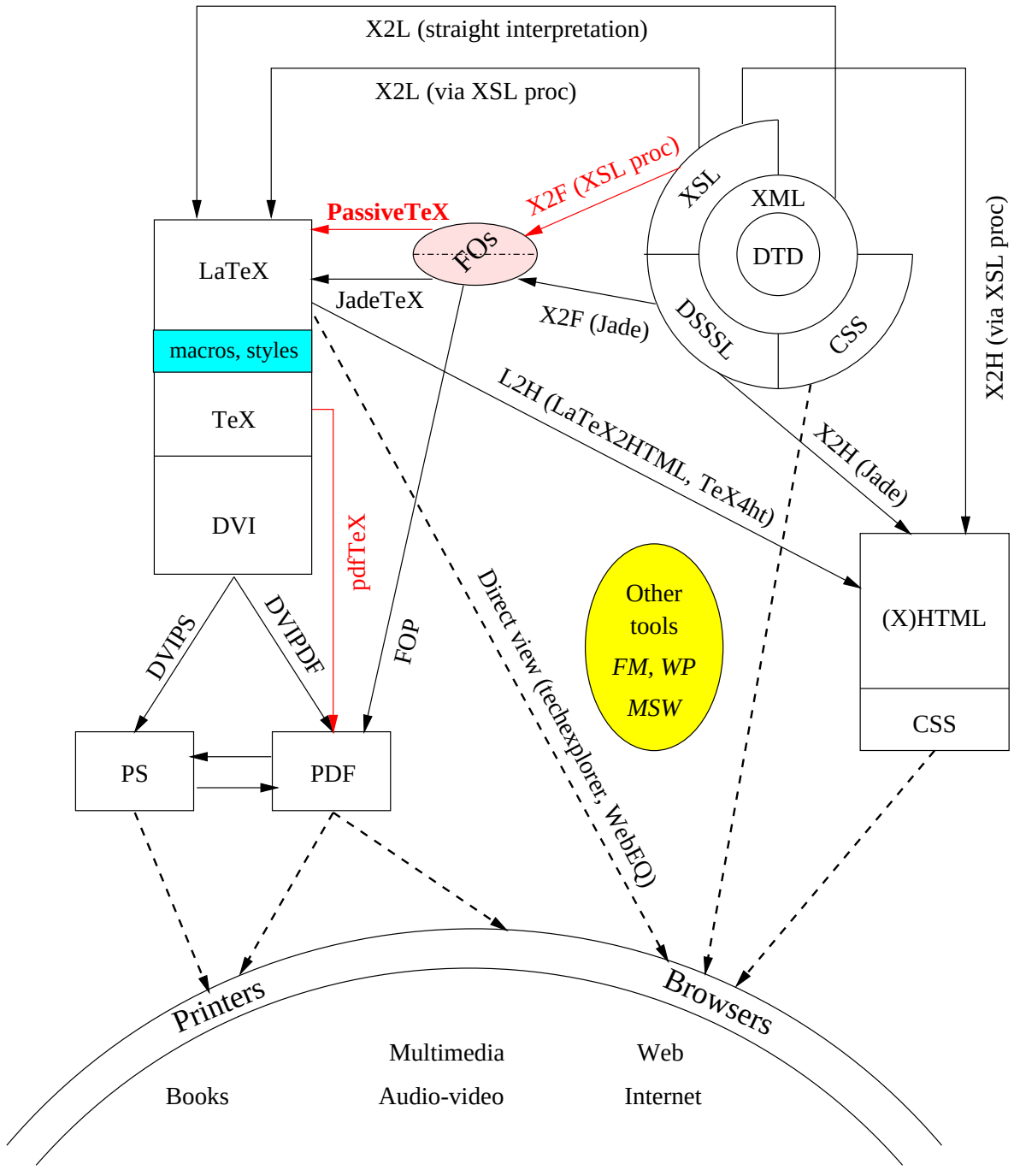
XML as central source repository

- The developers of XML have taken into account the lessons learned in the last decade using SGML and HTML.
- By construction XML is (should be) an ideal tool for dealing with most kinds of data and (multi-lingual) source documents (based on Unicode, seamless integration with modern languages, such as Java, perl, and python).
- Has gotten support from all corners of the Internet world: *Open Source* people as well as commercial players.
- Many free (and not-so-free) tools are available for all conceivable operating systems and purposes. Soon HTML-only browsers will only be a (bad) memory, and most Internet tools will support XML natively.

CERN

XML documents and the Internet

My (idealized) vision for the *not-too-far* future



CERN

Batch translation: `LaTeX2HTML`

- Written in Perl;
- uses LaTeX, `dvips`, `ghostscript`, `netpbm` utilities for the conversion of EPS images or math formulae into GIF or PNG;
- different hierarchical levels of source document put into different HTML files (under user control);
- table of contents, navigational aids;
- lot of packages supported.

CERN

Output example LaTeX2HTML

Math examples

$$\phi(\lambda) = \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} \exp(u\ln u + \lambda u)\,du \qquad \text{for } c \geq 0 \tag{1}$$

$$\lambda = \frac{\epsilon - \bar{\epsilon}}{\xi} - \gamma' - \beta^2 - \ln\frac{\xi}{E_{\max}} \tag{2}$$

$$\gamma = 0.577215\ldots \quad \text{(Euler's constant)} \tag{3}$$

$$\gamma' = 0.422784\ldots = 1 - \gamma \tag{4}$$

$$\epsilon, \bar{\epsilon} = \text{actual/average energy loss} \tag{5}$$

Since (6) or (7d) should hold for arbitrary $\delta\mathbf{c}$ -vectors, it is clear that $\mathcal{N}(A) = \mathcal{R}(B)$ and that when $y = F(x)$ one has...
...the Pythagorians knew infinitely many solutions in integers to $a^2 + b^2 = c^2$. That no non-trivial integer solutions exist for $a^n + b^n = c^n$ with integers $n > 2$ has long been suspected (Fermat, c.1637). Only during the current decade has this been proved (Wiles, 1995).

$$V\pi^{sr} = \left\langle \sum_i M_i \mathbf{V}_i \mathbf{V}_i + \sum_i \sum_{j>i} \mathbf{R}_{ij}\mathbf{F}_{ij} \right\rangle \tag{6}$$

$$= \left\langle \sum_i M_i \mathbf{V}_i \mathbf{V}_i + \sum_i \sum_{j>i} \sum_\alpha \sum_\beta \mathbf{r}_{i\alpha j\beta}\mathbf{f}_{i\alpha j\beta} - \sum_i \sum_\alpha \mathbf{p}_{i\alpha}\mathbf{f}_{i\alpha} \right\rangle$$

CERN

LaTeX2HTML – advantages

Quality

- High-level typography and performant navigational aids;
- cross-references intact (figures, tables, sections, equations);
- equations and figures (GIF) very readable;
- large series of user options to control translation process (“global” or “fractioned” translation for formulae);
- consistency in the visual presentation is an important design feature; (nevertheless some imprecise placement in the elements of the sub-formulae remains–also with `TeX4ht` and `tth`).

CERN

LaTeX2HTML – advantages (cont.)

Reliability and lisibility

- Runs without human intervention on thousands of systems worldwide (of course Perl must be installed also!);
- has no problem to handle very large documents (hundreds of pages) in a reliable way;
- LaTeX2HTML generates "standard" HTML files (3.2, 4.0);
- the "ALT" textual representation of the content of images permits an acceptable presentation of the information on non-graphic browsers, such as Lynx.

CERN

LaTeX2HTML – drawbacks

Execution time

- Can take several minutes for a few tens of pages;
- specially formulae and images take a lot of time;
- impossible to use for “on the fly” translation.

Diskspace

- Can easily generate hundreds of HTML files and GIF images;
- can double or triple the diskspace needed to store a document;
- maintenance and management non-trivial (export, regeneration).

CERN

LaTeX2HTML – Drawbacks (cont.)

Time to load the Web pages

- Large number of images to be transferred makes loading Web pages take a long time.

Installation

- `perl`, `ghostscript`, `netpbm` utilities, . . . must be installed (not so easy for a non-specialist);
- the complete LaTeX source document must be loaded into memory as a whole, so that a lot of system resources are needed during the translation process (can lead to problems on small systems, such as DOS).

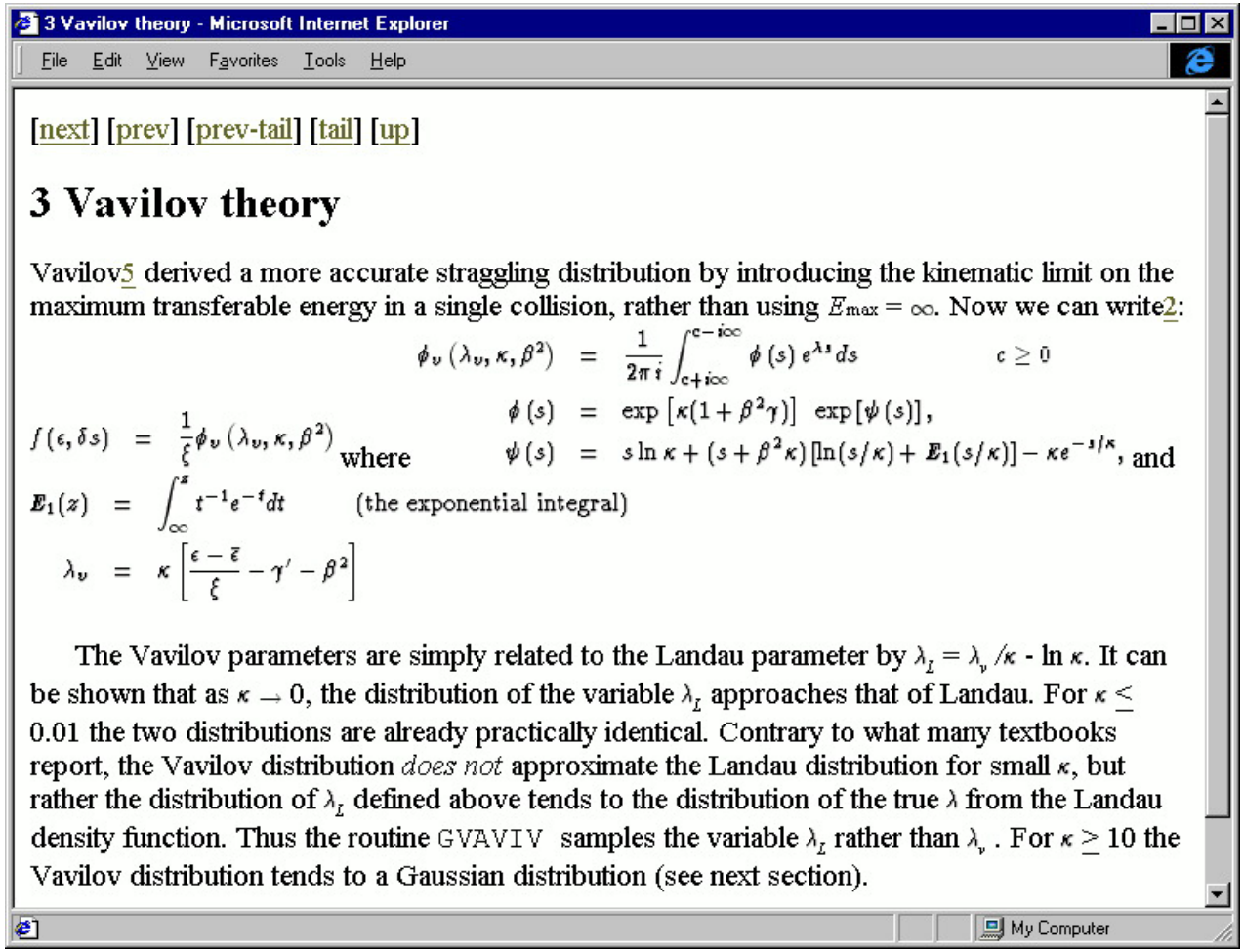
CERN

Batch translation *bis*: `TeX4ht`

- `TeX4ht` uses TEX itself to transform TEX commands into HTML tags using a dedicated LATEX package;
- does not need an external program, such as `perl`;
- uses the utilities of the `ghostscript` system to translate EPS files and mathematics formulae with special symbols;
- some additions in the LATEX sources might be necessary (to give some directives to the program).

CERN

Fragment generated by TeX4ht



CERN

TeX4ht – Advantages and drawbacks

- Mostly the same problems and successes as LaTeX2HTML;
- TeX4ht uses LaTeX with a few packages (for instance TeX4ht.sty) and some font files, making the installation a lot easier (especially on non-Unix systems, which do not have Perl and the other tools pre-installed);
- still needs dvips and ghostscript;
- compared to LaTeX2HTML the quality can be less good (e.g., placement of the limits on the integrals);
- may need modification in the source files.

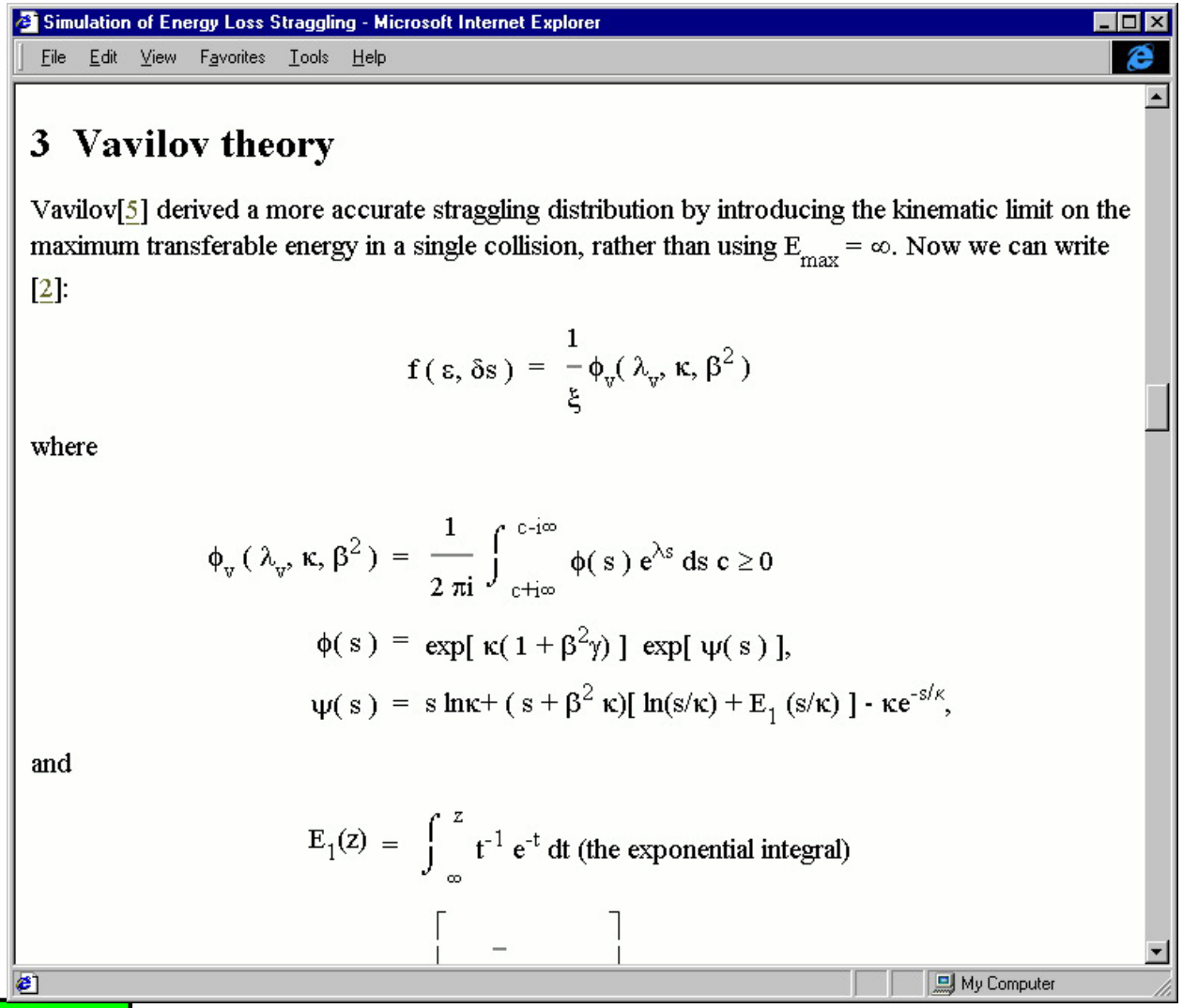
CERN

On the fly conversion: `tth`

- `tth` is written in C and uses no external programs, thus making the program extremely portable;
- formulae translated into HTML;
- uses the `Symbol` font available on X-Window, Mac ou PC;
- by default, images are not converted, but they are made available through external links (it is also possible to ask that GIF Images be generated, but that makes the procedure a lot more complicated).

CERN

Fragment generated by `tth`



CERN

`tth` – advantages

Performance acceptable

- Very fast (one pass over the data);
- source documents can be stored in a directory in compressed form (e.g., `.tar.gz`);
- the translation (`gtar zxvf`, then `tth`) can be performed very fast on the fly without any delay.

Easy maintenance

- Modifications must only be introduced in the sources which do not need to be retranslated.

CERN

tth – drawbacks

Medium quality

- Documents with only text or simple math come out not too bad;
- complete mess for complex formulae;
- (almost) no cross-references.

Modifications in the working environment

- The user's working visualisation environment must be adapted (X-window files, program setup), otherwise the documents will be unreadable.

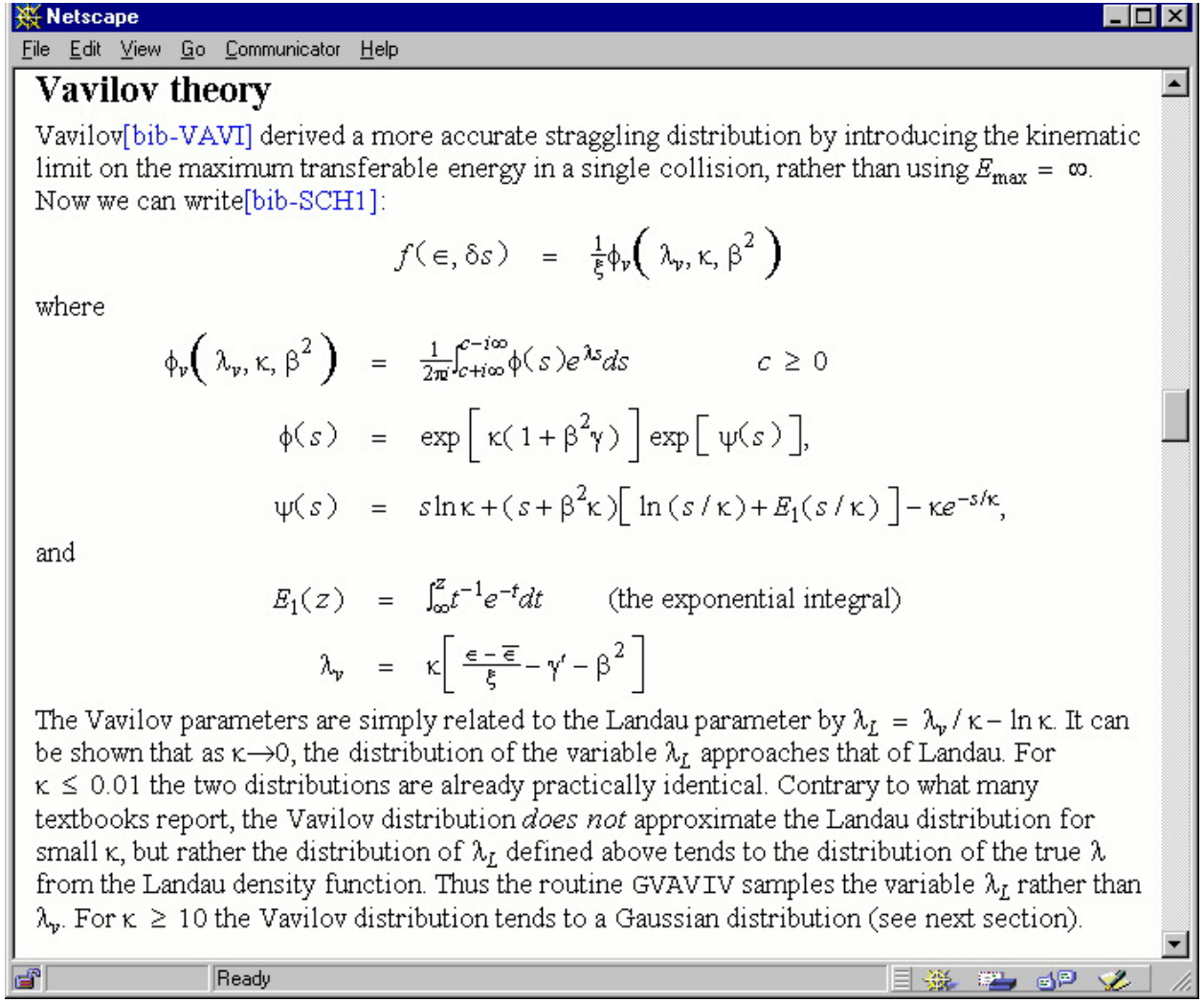
CERN

“Online” LaTeX: `techexplorer`

- Extension module (*plug-in*) pour Netscape (NS) and MS Internet Explorer (MSIE);
- (free) version (*Introductory Edition*) is available for W95/NT, SGI IRIS 6.2, Sun Solaris 2.5 and IBM AIX 4.1;
- `techexplorer` reads directly the LaTeX source document;
- does not (yet) handle all LaTeX commands (uninterpreted commands are displayed *as-is* in red in the browser);
- `techexplorer` offers a whole set of hypertext and multimedia extensions;
- attractive tool for preparing documents for and on the Web;
- initial support for MathML exists.

CERN

Fragment displayed with `techexplorer`



CERN

Equation browsing with Java and WebEQ

- Interprets WebTeX (subset of LaTeX);
- formulae input in TeX-syntax between $...$ or \[...\];
- special utility (the *Sizer*) translates them into Java applets that can be visualized with WebEQ;
- WebEQ has its own editor, which has support for direct MathML input;
- the WebEQ editor is an ideal teaching tool for learning about MathML since it allows you to translate (not too complicated) formulae from LaTeX into MathML.

CERN

Fragment generated by WebEQ

$$\phi_\nu\left(\lambda_\nu,\kappa,\beta^2\right)=\frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty}\phi(s)e^{\lambda s}ds \qquad c\geq 0$$

$$\phi(s)=\exp\left[\kappa(1+\beta^2\gamma)\right]\exp\left[\psi(s)\right],$$

$$\psi(s)=s\ln\kappa+(s+\beta^2\kappa)\left[\ln(s/\kappa)+E_1(s/\kappa)\right]-\kappa e^{-s/\kappa},$$

and

$$E_1(z)=\int_z^\infty t^{-1}e^{-t}dt \qquad \text{(the exponential integral)}$$

$$\lambda_\nu=\kappa\left[\frac{\epsilon-\bar{\epsilon}}{\xi}-\gamma'-\beta^2\right]$$

The Vavilov parameters are simply related to the Landau parameter by $\lambda_L=\lambda_\nu/\kappa-\ln$. It can be shown that as $\kappa\to\mathrm{C}$, the distribution of the variable λ_L approaches that of Landau. For $\kappa\leq 0.0$ the two distributions

CERN

TEX brings typography to XML

- Often typographic quality is a *must*.
- The *print* button of most present-day HTML browsers does not in general give high quality printable copy.
- *Special* and separate procedures are used to prepare the printable output for XML documents.
- With SGML and DSSSL James Clark's Jade (nowadays being extended by members of the DSSSL community as open source library under the name of Openjade, see `http://jade-cvs.avionitek.com/`).
- Jade reads DSSSL style sheets and provides several back ends (FOT, TEX, *ML, RTF).
- Sebastian Rahtz' (David Megginson) `jadetex` generates a printable document (PostScript, PDF).

CERN

What is PassiveTEX?

- PassiveTEX is a library of TEX macros which can be used to process an XML document resulting from an XSL transformation to formatting objects.
- PassiveTEX provides a rapid development environment for experimenting with XSL FO, using a reliable pre-existing formatter.
- Running PassiveTEX with the pdfTEX variant of TEX generates high-quality PDF files in a single operation.
- PassiveTEX shows how TEX can remain the formatter of choice for XML, while hiding the details of its operation from the user.

PassiveTEX is available at
`http://users.ox.ac.uk/~rahtz/passivetex/`.

CERN

PassiveTEX: history and components

PassiveTEX derives from and builds on:

- `typehtml`, a LATEX package by David Carlisle, used to typeset HTML directly using LATEX;
- `jadetex`, a package by Sebastian Rahtz that implements the output of the Jade DSSSL processor's TEX backend;
- A UTF-8 handler by David Carlisle, in conjunction with the catalogue of Unicode/TEX mappings built up for `jadetex`.

The system components include macros to:

- parse XML input (attributes, entities, etc.), parse UTF-8 input, and MathML elements;
- instantiate formatting object elements and attributes;
- map Unicode to TEX font layouts.

CERN

Advantages and disadvantages of PassiveTEX

- ☺ rapid development;
- ☺ well-understood, robust, stable, and freely available page formatter;
- ☺ fonts, graphics inclusion, hyperlinks, etc. come for free;
- ☺ mature handling of language issues, including hyphenation;
- ☺ high-quality math rendering (TEX's *raison d'être*);
- ☺ pdfTEX variant generates very high-quality PDF.

- ☹ constraint to use TEX's page makeup model, and force XSL FO to fit it;
- ☹ as LATEX is already high-level markup, it is too easy to allow things to fall through and take LATEX defaults;
- ☹ TEX macro writing is obscure and difficult, so that the system is not transparent for most (non-TEX) programmers;
- ☹ TEX is large and monolithic (difficult to embed in other applications);
- ☹ TEX seems much like a sledgehammer to crack a nut . . .

CERN

PassiveTEX: how it handles maths

PassiveTEX supports MathML directly. An XSL style sheet can pass `<math>` and its children through unchanged, as follows:

```
<xsl:template match="math">
  <xsl:apply-templates mode="math"/>
</xsl:template>

<xsl:template mode="math"
  match="*|@*|comment()|processing-instruction()|text()">
  <xsl:copy>
    <xsl:apply-templates mode="math"
      select="*|@*|processing-instruction()|text()"/>
  </xsl:copy>
</xsl:template>
```

A reasonable subset of presentation MathML is recognized, and produces good output. We show an example later.

CERN

XSL, FO and TEX, cons and pros

- ☹ The XSL FO page model is inherited from DSSSL, and is unproven for production-quality print formatting.
- ☹ The XSL specification is unfinished and incomplete.
- ☹ One cannot easily tweak TEX's behaviour with this system.
- ☹ The table model of XSL is sufficiently far from TEX's that it may require a pre-processor.

- ☺ Toegther with FOP, we now have systems to experiment with, and commercial implementations cannot be far behind (can they?).
- ☺ TEX is close to being a XSL FO-capable formatter.
- ☺ With the Omega TEX variant (using Unicode internally), we have a native Unicode typesetting system ready and waiting.
- ☺ XSL FO does not threaten TeX — it gives it a reason to survive.

CERN

PassiveTEX, typesetting a scientific document

- original source is LaTeX;
- translated to XML with TeX4ht using LaTeX-like *ad hoc* DTD and MathML;
- write an XSL style sheet to treat the "textual elements" of the document;
- "pass through" the math components to the back-end; (XSL, and DSSSL up to a point, do not have a sufficient set of FOs to do math correctly, and although the X3C Math WG is talking to the XSL WG, at present the present consensus seems to be that it is best to treat the MathML directly at the end application level);
- interpret MathML directly in PassiveTeX.

$$\text{LaTeX} \xrightarrow{\text{TeX4ht}} \text{XML} \xrightarrow{\text{XSL}} \text{FO and MathML} \xrightarrow{\text{PassiveTeX}} \text{PDF}$$

CERN

The generated MathML source

```
<section id="vavref">
<stitle>Vavilov theory</stitle>

<par>Vavilov<cite refid="bib-VAVI"/> derived a more accurate
straggling distribution by introducing the kinematic limit on the
maximum transferable energy in a single collision, rather than using
<inlinemath><math><msub><mi>E</mi><mrow><mtext>max</mtext></mrow></msub>
<mo>=</mo><mi>&infin;</mi></math></inlinemath>.

Now we can write<cite refid="bib-SCH1"/>:

<eqnarray ><subeqn><math><mi>f</mi> <mfenced open='(' close=')'>
<mi>&epsi;</mi><mo>,</mo><mi>&delta;</mi><mi>s</mi></mfenced>
<mo>=</mo> <mfrac><mrow><mn>1</mn></mrow>
<mrow><mi>&xi;</mi></mrow>
</mfrac><msub><mi>&phi;</mi><mrow><mi>v</mi></mrow></msub>
<mfenced open='(' close=')'>
<msub><mi>&lambda;</mi><mrow><mi>v</mi></mrow></msub><mo>,</mo>
<mi>&kappa;</mi><mo>,</mo><msup><mi>&beta;</mi><mrow><mn>2</mn></mrow>
</msup></mfenced></math></subeqn></eqnarray>
where
<eqnarray><subeqn><math><msub><mi>&phi;</mi><mrow><mi>v</mi></mrow></msub>
```

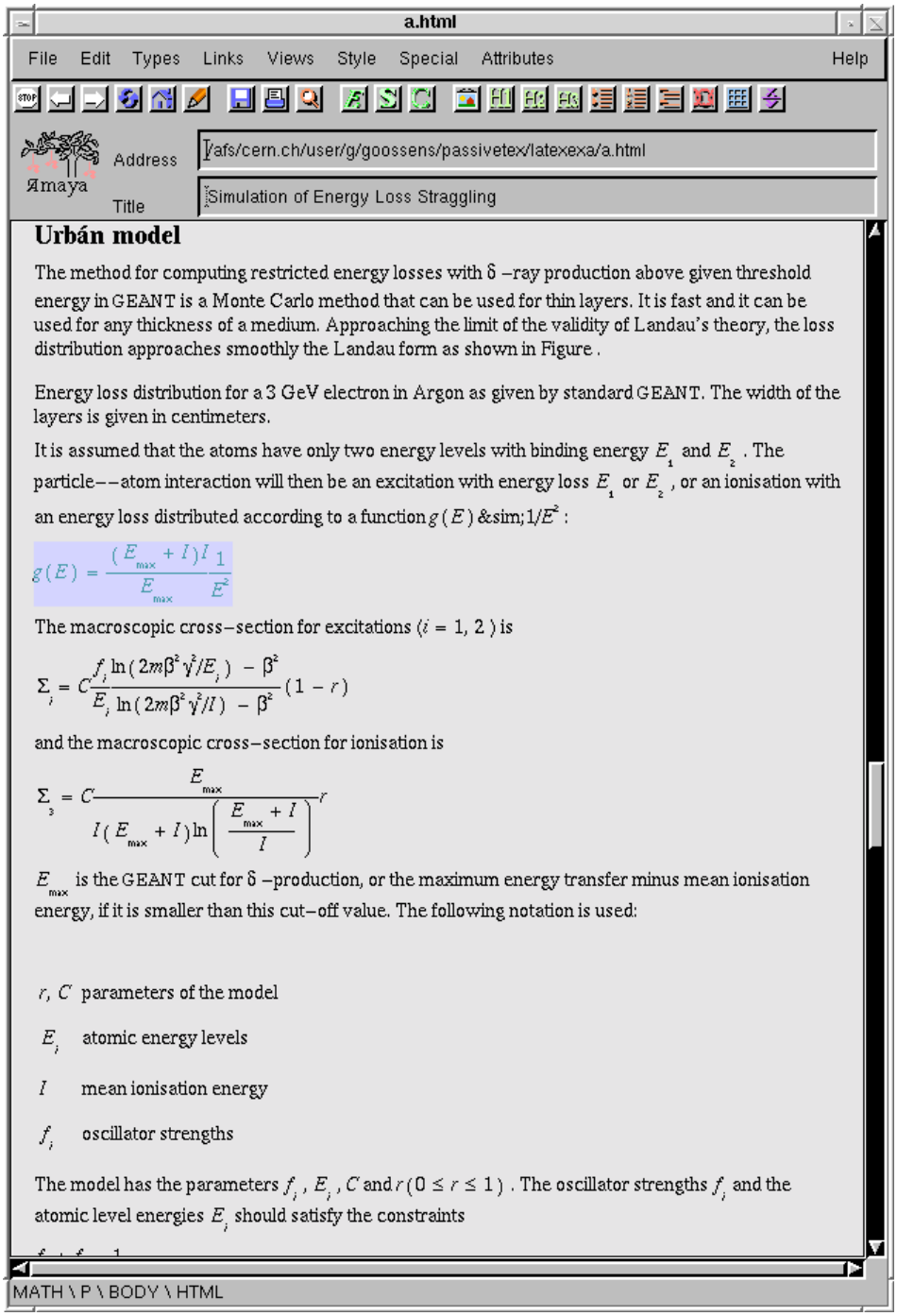
CERN

```
<mfenced open='(' close=')'>
<msub><mi>&lambda;</mi><mrow><mi>v</mi></mrow></msub><mo>,</mo>
<mi>&kappa;</mi><mo>,</mo>
<msup><mi>&beta;</mi><mrow><mn>2</mn></mrow></msup></mfenced>
  <mo>=</mo>
<mfrac><mrow><mn>1</mn></mrow>
       <mrow><mn>2</mn><mi>&pi;</mi><mi>i</mi></mrow>
</mfrac>
<msubsup><mo>&int;</mo>
<mrow><mi>c</mi><mo>+</mo><mi>i</mi><mi>&infin;</mi></mrow>
<mrow><mi>c</mi><mo>-</mo><mi>i</mi><mi>&infin;</mi></mrow></msubsup>
<mi>&phi;</mi><mfenced open='(' close=')'><mi>s</mi></mfenced>
<msup><mi>e</mi><mrow><mi>&lambda;</mi><mi>s</mi></mrow></msup>
<mi>d</mi><mi>s</mi><mspace width='2cm'/><mi>c</mi><mo>&geq;</mo><mn>0</mn>
                 </math></subeqn>

<subeqn><math><mi>&phi;</mi><mfenced open='(' close=')'><mi>s</mi></mfenced>
<mo>=</mo><mo>exp</mo><mfenced open='[' close=']'><mi>&kappa;</mi>
<mrow><mo>(</mo><mn>1</mn><mo>+</mo><msup><mi>&beta;</mi>
      <mrow><mn>2</mn></mrow></msup><mi>&gamma;</mi><mo>)</mo></mrow>
</mfenced><mo>exp</mo><mfenced open='[' close=']'><mi>&psi;</mi>
<mfenced open='(' close=')'><mi>s</mi></mfenced></mfenced>
<mo>,</mo> </math></subeqn>
```

CERN

MathML viewed with Amaya



CERN

$$f(\epsilon;\delta x) \approx \frac{1}{\xi\sqrt{\frac{2\pi}{\kappa}(1-\beta^2/2)}} \exp\left[\frac{\lambda - \bar{\epsilon}}{2} - \frac{e^{\lambda}}{\xi^2(1-\beta^2/2)}\right]$$

thus implying

$$\text{mean} = \bar{\epsilon}$$

$$\sigma^2 = \frac{\xi}{\kappa}(1-\beta^2/2) \qquad \xi E_{\max}(1-\beta^2/2)$$

5 Urbán model

The method for computing restricted energy losses with δ-ray production above given threshold energy in `GEANT` is a Monte Carlo method that can be used for thin layers. It is fast and it can be used for any thickness of a medium. Approaching the limit of the validity of Landau's theory, the loss distribution approaches smoothly the Landau form as shown in Figure 2.

It is assumed that the atoms have only two energy levels with binding energy E_1 and E_2. The particle–atom interaction will then be an excitation with energy loss E_1 or E_2, or an ionisation with an energy loss distributed according to a function $g(E) \sim 1/E^2$:

$$g(E) = \frac{(E_{\max} + I)I}{E_{\max}} \frac{1}{E^2} \tag{1}$$

The macroscopic cross-section for excitations $(i = 1, 2)$ is

$$\Sigma_i = C \frac{f_i}{E_i} \frac{\ln(2m\beta^2\gamma^2/E_i) - \beta^2}{\ln(2m\beta^2\gamma^2/I) - \beta^2}(1 - r) \tag{2}$$

General conclusion

- XML allows you to construct your own markup language, optimized for the task at hand and using tags and attribute names in the user's native language and an encoding adapted to the local computer system.
- A lot of tools are freely (and not so freely) available.
- All major Internet players have adopted the standard.
- "Standard" DTDs are already being prepared by various scientific, commercial, etc., communities, and this will improve communication and interoperability
- XSL is a good companion tool for styling and transforming XML sources. It is well integrated with the Web and builds on DSSSL and CSS (formatting objects, SVG,...).
- In "our" area, preparing scientific documents for the Web is a non trivial task, especially if they contain mathematical information that we want to "share".

CERN

- We have looked at a few “non-optimal” solutions: PDF/PostScript (Java, plugins) or GIF images.
- We think/hope that XML, MathML and companion specifications (SVG, XML Schema, etc.) will allow us to handling scientific documents on the Web gracefully.
- TEX is a reliable output engine that can with rather minimal work provide typographically excellent output, especially for typesetting scientific XML documents containing a lot of maths;
- With well-known and “standard” DTDs (TEI, DocBook, ISO12083, etc.) XML can be used as a *lingua franca* to transport documents between various editing and document handling systems. At CERN we plan to test portability between XML, LATEX, and FrameMaker/SGML; later also Microsoft Word, Wordperfect, etc. when these applications will have better support for XML and XSL.
- An *Esprit* project TIPS (*Tools for Innovative Publishing in Science*) will look at possible solutions.

CERN