

# CESNET Technical Report 1/2010

## XML Markup Language for Technical Reports

LADISLAV LHOTKA

*CESNET, z.s.p.o.*

Received 15.2.2010

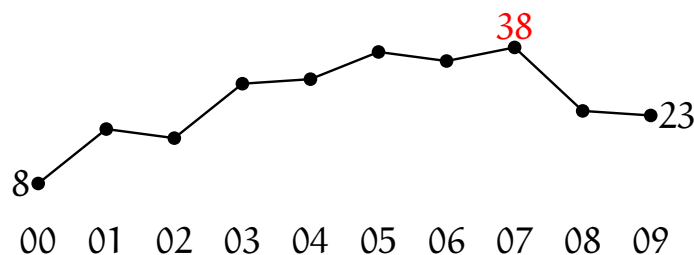
### Abstract

This technical report describes the second version of Techrep XML markup language which is primarily intended for preparing source text of technical reports published by CESNET. Techrep2 also serves as the common internal format to which all other formats (ODT, L<sup>A</sup>T<sub>E</sub>X, DocBook, reStructuredText and Techrep1) are translated before further processing. Techrep2 retains the simplicity of the original version but consolidates the markup language in several important ways. In particular, Techrep2 vocabulary now belongs to an XML namespace, which allows for combinations with other vocabularies in the future. Based on the experience with Techrep1, this version also introduces a limited number of new XML elements and attributes for frequently used text structures.

*Keywords:* XML, markup language, technical reports

## 1 Introduction

CESNET technical reports have been published since 2000 as a medium for easy, rapid and broad dissemination of research and development results. Over the period of 10 years, 264 technical reports were published (see Figure 1) covering all areas of research and development carried out by CESNET and its project partners.



**Figure 1.** Annual counts of published technical reports.

The very first technical report [11] described the original Techrep version 1, which was, at that time, the only accepted format of CESNET technical reports. The leading design principle of Techrep v1 was simplicity. The XML language had its strong footing in HTML that most potential authors were supposed to be familiar with. Moreover, the simplicity of the XML language aided the development of transformation scripts generating the two target formats – HTML and PDF (via L<sup>A</sup>T<sub>E</sub>X).

However, the Techrep markup language never gained much popularity. By and large, this was due to the fact that most authors were not able or willing to

prepare the source text directly in *any* XML language. On the other hand, XML enthusiasts were discouraged by the apparent shortcomings of Techrep compared to, for example, DocBook [12]. As a result, more and more technical reports were prepared in other formats that could not be processed into an uniform presentation style.

In 2006, CESNET decided to start publishing printed annual proceedings of selected technical reports. Therefore, it became necessary to be able to render all technical reports in the same visual style. At the same time, it was quite clear that many authors were strongly attached to (or spoiled by) the mainstream methods of document preparation – mainly MS Word and L<sup>A</sup>T<sub>E</sub>X – and didn't want to consider XML as an option. Finally, a compromise was found: potential authors can now submit their technical reports in multiple formats that are eventually translated into a common internal form, which is Techrep version 2. The accepted formats are: Open Document Text [4], L<sup>A</sup>T<sub>E</sub>X [8], DocBook [12], reStructuredText [5], Techrep1 [11] and, naturally, Techrep2, which is described in this report.

In order to fit well into the role of *lingua franca* for the above formats and thus both source and destination for XSLT transforms, the main criterion in designing Techrep2 was simplicity and flexibility. Despite of that, or perhaps because of that, the result is still quite suitable as the source format for authors who are used to writing XML and HTML in a text editor. Arguably, Emacs<sup>1</sup> with nXML<sup>2</sup> mode is one of the fastest and most effective ways of preparing XML text in general, and technical reports in particular.

The main changes in Techrep version 2 compared to version 1 are:

1. use of XML namespace;
2. English element names;
3. better overall structure;
4. clear distinction between block and inline elements;
5. a special element for intra-document cross-references;
6. standard common attributes such as `xml:id`.

A number of XSLT stylesheets are readily available for transforming the accepted formats to Techrep2 and then, in turn, producing XHTML and T<sub>E</sub>X. These tools are described in Section 9.

Appendix A contains the formal definition of the Techrep2 language expressed as a RELAX NG schema [7]. For easy reference, Appendix B provides an alphabetically sorted table of all Techrep2 element tags.

In order to provide an extended example, the Techrep2 source of this technical report is also available online<sup>3</sup>.

---

<sup>1</sup> <http://www.gnu.org/software/emacs>

<sup>2</sup> <http://www.thaiopensource.com/nxml-mode>

<sup>3</sup> <http://www.cesnet.cz/doc/techzpravy/2010/techrep2/techrep2.ctr>

## 2 Overall Structure of the Source Text

Techrep2 documents are usually stored as disk files containing the XML source. The recommended file name extension is `.ctr`.

All XML elements belong to an XML namespace [2] represented by URL

```
http://cesnet.cz/ns/techrep/base/2.0
```

The overall structure of a technical report is as follows:

```
<report xmlns="http://cesnet.cz/ns/techrep/base/2.0">
  ... metadata ...
  <body>
    ... main text ...
  </body>
</report>
```

The root element is `<report>`. Apart from the common attributes (see Section 3), this element may have the following optional attributes:

**number**

Serial number of the technical report, allocated by the editor.

**status**

Status of the report, its value may be either `proof` (default) or `final`.

The `<body>` element contains the main text body and separates it from the front matter (metadata). The main text body consists of block elements (Section 6) and sectioning elements (Section 5).

## 3 Common Attributes

The following two attributes may be attached to any element:

**xml:lang**

Language used in the contents of the element. Its value is a two-letter language code from the IANA Language Subtag Registry<sup>4</sup>, see [10]. The default value is `en` for English, `cs` is used for Czech. The language selection may be locally overridden by using the `xml:lang` attribute on a descendant element.

**role**

User-defined role of the element. This attribute may be used for extending the Techrep2 vocabulary by defining special semantics, which may be reflected in output rendering.

Another common attribute is `xml:id`, which defines a unique identifier for an element, see [9]. It is used in particular for specifying a target for cross references and bibliographic citations. Its value must be of type ID [3], in particular, it must start with a letter, underline or colon. The `xml:id` attribute is mandatory for `<affiliation>` and `<bibitem>` elements and may be attached to the following elements, whenever they need to serve as targets for cross references: `<report>`, `<h1>`, `<h2>`, `<h3>`, `<appendix>`, `<figure>`, `<table>`, `<ol>`, `<li>` and `<dt>`.

<sup>4</sup> <http://www.iana.org/assignments/language-subtag-registry>

## 4 Metadata

The document element `<report>` is immediately followed by several mandatory elements containing metadata about the technical report. For example, this report has the following metadata:

```
<title>XML Markup Language for Technical Reports</title>
<authors>
  <author affil="CESNET"
    email="lhotka@cesnet.cz">Ladislav Lhotka</author>
  <affiliation xml:id="CESNET">CESNET, z.s.p.o.</affiliation>
</authors>
<date>15.2.2010</date>
<abstract>...</abstract>
<keywords>...</keywords>
```

### 4.1 The `<title>` Element

This element contains the title of the technical report. It is mandatory and must always come as the first child of the `<report>` element. Inline markup is allowed in the content but should be used very sparingly.

If the title is long, the optional attribute `short` may be used to specify a shorter version of the title, which is then used in page headings etc.

### 4.2 The `<authors>` Element

Immediately following the `<title>` element is the mandatory `<authors>` element containing information about the authors in its subelements:

- At least one `<author>` element containing an author's name must be present. It is not further structured and should be given in the order *first name*, *middle initials* (if any) and *family name*. The `<author>` element has two optional attributes: `email` should contain author's email and `affil` points to an affiliation via the ID reference mechanism.
- Optional `<affiliation>` element(s) contain one or more affiliations of the authors. Every `<affiliation>` element must have the `xml:id` attribute containing the unique ID that serves as a target for the `affil` attribute (see above).
- Optional `<email>` element contains author's email address.

### 4.3 The `<date>` element

This element is optional and should contain the date when the report was first submitted. Its format is not enforced but should be in the form *DD.MM.YYYY*, for example "15.02.2010".

#### 4.4 The <abstract> element

This mandatory element contains the abstract of the technical report. It is one of the elements with the hybrid content model (see Section 6) so that it may be written using either inline markup only or, alternatively, one or more block elements.

#### 4.5 The <keywords> element

This element contains a comma-separated list of key words or phrases. It is mandatory and may either precede or follow the <abstract> element. No markup is allowed in the content.

### 5 Sectioning Elements

Techrep2 retains the HTML style of text structuring with three levels of section hierarchy represented by elements <h1>, <h2> and <h3>. These elements only demarcate the beginning of a section, which is otherwise not encapsulated. While this flat design is considered inferior to nested sectioning structures, it was used in Techrep2 because the simplicity and resemblance to HTML should make it more accessible to potential authors.

Unlike HTML where sectioning elements are often use quite loosely, Techrep2 documents must strictly follow the *logical* hierarchy of sections. This means that top-level sections must use <h1>, <h2> sections must be logically inside <h1> sections and similarly for <h3>. Note that in contrast to the HTML practice, <h1> elements are used for top-level sections of the report and not for the report title.

Techrep2 introduces a new sectioning element for appendices – <appendix>. Like <h1>, this element represents the highest section level and the text of an appendix may be further subdivided by <h2> and <h3> sections. However, <appendix> is only allowed to appear *after* the special bibliography section (see Section 8) whereas all <h1> sections must precede the bibliography.

Sectioning elements <h1>, <h2>, <h3> and <appendix> contain the section title, which may use inline markup. In addition, they may have any of the common attributes (Section 3), of which the most useful is the `xml:id` attribute that makes the section into a target for cross references (see Section 7.1).

### 6 Block Elements

Block elements are paragraphs or other essentially two-dimensional objects. In most cases they have inline contents, which means text and inline elements (see Section 7). Certain block elements may contain other block elements while others allow *hybrid content* – either mixed content with inline elements or other block elements.

Unless otherwise noted, common attributes (Section 3) may be added to all block elements.

## 6.1 Paragraph

Paragraphs are basic text structuring units. Every paragraph is contained inside a single `<p>` element. With one exception, paragraphs only allow inline contents but not other block elements. In particular, figures and tables must always appear outside paragraphs. The exception is the *compact list* which may also appear inside paragraphs (see Section 6.4 for more details).

## 6.2 Block Quote

Longer stretches of quoted text, program listings and similar objects may be enclosed by the `<blockquote>` element. This results in special rendering, typically with increased indentation. The `<blockquote>` element may only contain other block elements.

## 6.3 Preformatted Block

Any text enclosed by the `<pre>` element is protected from output formatting. This is useful for program listings, configuration files etc. Note that the input text must still be valid XML so that, for instance, “<” characters must be properly quoted. Therefore, it is often useful to utilise CDATA blocks for such purposes.

## 6.4 Lists

Techrep2 provides lists in three variants that closely follow the HTML model: ordered, unordered and definition lists.

*Ordered lists* are enclosed by `<ol>` elements and contain any number of `<li>` elements representing list items.

List items are labelled with an ordered sequence of numbers or letters. The style of labels is controlled by the optional `labels` attribute attached to the `<ol>` element. Its value may be one of the following choices:

- `arabic` – Arabic numerals 1, 2, 3, ... (this is the default);
- `roman` – lowercase Roman numerals i, ii, iii, ...;
- `ROMAN` – uppercase Roman numerals I, II, III, ...;
- `alpha` – lowercase letters a, b, c, ...;
- `ALPHA` – uppercase letters a, b, c, ...;

Another optional attribute of the `<ol>` element – `continue` – allows for continuing a previous ordered list. This is useful when one wants to temporarily suspend an ordered list, insert one or more paragraphs or other block elements outside the list and then continue with the next list item in sequence. To achieve this, the `<ol>` element enclosing the first part of the list must be given a unique ID in the `xml:id` attribute, which is then used as the value of the `continue` attribute. This way, a single list may be suspended and resumed multiple times and its successive parts are joined in a chain-like manner via the `xml:id` and `continue` attributes.

*Unordered lists*, enclosed by `<ul>` elements, are similar to their ordered counterparts except that their labels are identical symbols such as bullets or dashes. Unlike ordered lists, the symbols are selected automatically based on the level of the unordered list.

Both ordered and unordered lists have a special *compact* variant. It is intended for lists with single-line items where the vertical spacing between items may be reduced. Items of compact lists must not contain block elements and cannot be continued. Every list appearing inside a paragraph is automatically compact. Otherwise, <ol> or <ul> lists appearing outside paragraphs can be made compact by adding the attribute `compact` with the value `true` to the <ol> or <ul> element. For lists inside paragraphs, the `compact` attribute is not allowed.

Items in non-compact <ol> or <ul> lists use the hybrid content model, that is either mixed content with inline elements or block elements only. For example, the first item in the following unordered list contains two paragraphs while the second item uses mixed content with inline elements:

```
<ul>
  <li>
    <p>First paragraph of the first item.</p>
    <p>Second paragraph of the first item.</p>
  </li>
  <li>Second item with an <em>emphasised text</em>.</li>
</ul>
```

The third type of list is the *definition list* enclosed by <dl>. Every item in a definition list consists of one or more <dt> elements – definition term(s) – followed by exactly one <dd> element – description. Note that, unlike HTML, Techrep2 allows multiple terms corresponding to one description, for example

```
<dl>
  <dt>Firefox</dt>
  <dt>Internet Explorer</dt>
  <dt>Opera</dt>
  <dd>Popular graphical web browsers.</dd>
  <dt>Links</dt>
  <dt>Lynx</dt>
  <dd>Text-based web browsers.</dd>
</dl>
```

## 6.5 Logical Figures and Tables

Elements <figure> and <table> represent two independent types of serially numbered objects. They may only appear at the top level of the document body, that is, at the same level in as the sectioning elements <h1>, <h2> and <h3>.

Both <figure> and <table> must contain one or more block subelements. As the names suggest, <figure> and <table> elements usually contain an image (Section 6.6) or tabular object (Section 6.7), respectively, but this is by no means necessary.

Both <figure> and <table> have another mandatory subelement <caption>, which contains the figure or table caption. The <caption> element uses the hybrid

content model, although in most cases only mixed content with inline elements is expected. The `<caption>` element may appear in the XML source before or after the other subelements of `<figure>` or `<table>`.

Note that numbers of figures and tables are not given in the XML source – they are supposed to be added automatically by the processing software. Another point worth mentioning is that in a paginated output such as  $\text{\TeX}$ , both `<figure>` and `<table>` may become *floating* objects. Therefore, authors must be prepared that in the typeset output figures and tables needn't be at the same place as in the input but may overflow to a subsequent page.

## 6.6 Image

The `<image>` element is a container for graphical objects such as graphs or photographs. Unlike the first version of Techrep that relied on certain file naming convention, Techrep2 allows to specify multiple graphical formats for the same image stored in files with arbitrary names: the `<image>` element must have one or more `<source>` subelements containing references to such files.

The `<source>` element has two mandatory attributes:

`format`

This attribute specifies the graphical format using one of the following values: EPS, GIF, JPEG, PDF, PNG and SVG.

`file`

This attribute gives an uniform resource locator (URL), typically a local file name, which points to the graphics source file.

For example, Figure 1 above was specified as follows:

```
<figure xml:id="fig-pocty">
  <image>
    <source format="PDF" file="pocty.pdf"/>
    <source format="PNG" file="pocty.png"/>
  </image>
  <caption>Annual counts of published technical reports.</caption>
</figure>
```

References to an image should be addressed to the enclosing `<figure>` element. It is nevertheless allowed to use an `<image>` element without the `<figure>` wrapper, but this image will *not* become a floating object when the document is typeset. If it doesn't fit on the page at the place where it is specified, the page layout will be broken.

## 6.7 Tabular Objects

The `<tabular>` element represents a rectangular table of objects. Its content model resembles HTML tables. The `<tabular>` element has one mandatory attribute, `colspec`, and one or more subelements `<tr>` representing table rows. The value of the `colspec` attribute is a character string whose length must be equal to the

number of table columns. Each character specifies the horizontal alignment of one table column: “l” means left-aligned, “r” right-aligned and “c” centred.

The `<tr>` element may contain any combination of `<td>` and `<th>` elements – their total number is determined by the `colspec` attribute of the parent `<tabular>` element. The `<td>` element represents a normal table cell, whereas `<th>` is a header cell, which is usually rendered differently in the output.

The `<tr>` element may also have an optional `bgcolor` attribute that may be used to specify the background colour of the table row. Its value is the hash mark “#” followed by six hexadecimal digits defining the RGB value. The default value of this attribute is “#FFFFFF” (white).

The `<td>` and `<th>` use the hybrid content model. This means, for example, that a table can be used for arranging an array of images. In addition, `<td>` and `<th>` elements may have the following three optional attributes:

#### `bgcolor`

This attribute defines the background colour for the table cell. It is used in exactly the same way as with `<tr>` elements.

#### `align`

This attribute allows to override the horizontal alignment specified for the current column by the `colspec` attribute of `<tabular>`. Permitted values are `left`, `center` and `right`.

#### `colspan`

This attribute contains a natural number which says that the cell content should span across that many columns.

## 7 Inline Elements

Together with simple text, inline elements form the contents of some block elements, typically paragraphs or list items. The most frequent use of inline elements is for defining emphasis, font type selection and similar purposes:

#### `<em>`

normal emphasis, usually rendered with *italics*;

#### `<strong>`

strong emphasis, usually rendered with **boldface**;

#### `<tt>`

monospaced (typewriter) font;

#### `<sup>`

superscript, upper index, as in  $x^2$ ;

#### `<sub>`

subscript, lower index, as in  $x_2$ ;

`<q>`

quoted text with double quotes appropriate for the currently selected language – for example, `<q xml:lang="en">English</q>` results in “English” and `<q xml:lang="cs">české</q>` in „české“.

Other inline elements identify logical text objects:

`<command>`

operating system command

`<file>`

file name

`<input>`

data entered by the user

`<uri>`

Uniform Resource Identifier [1].

The remaining three inline elements are somewhat special:

`<br/>`

This element, which must always be empty, causes a line break at the point of the text where it appears.

`<footnote>`

This element specifies a footnote and allows for hybrid content.

`<phrase>`

This element offers the possibility of extending the choice of inline elements, especially in connection with the `role` attribute, for example

```
<phrase role="xml-elem">report</phrase>
```

## 7.1 Cross References

Cross-referencing elements serve in the text for referring to objects within the same document (sections, figures, tables, bibliography items etc.), but also to external resources.

The `<xref>` element is used for intra-document references. The value of its mandatory attribute `linkend` must be the ID (value of the `xml:id` attribute) of the element that is referred to. The other attribute of `<xref>` is `raw`. It is optional and controls the rendering of the reference in the output. If its value is “false”, which is the default, the referring text consists of the name and number of the referred object, for example “Section 7.1”. If `raw` is “true”, only the number is used: “7.1”.

In most cases, the `<xref>` element is empty, but it may also have inline content. In this case, the inline content appears first and then comes the cross-reference text enclosed in parentheses. For online output such as XHTML, the inline content becomes also part of the “hot link”. For example,

```
<xref linkend="sec-xref">hyperlink</xref>
```

is rendered this way: [hyperlink](#) (Section 7.1).

References to items in the bibliography list (Section 8) use the `<cite>` element. It is always empty and has one mandatory attribute `bibref` with the ID of the bibliographic item.

References to external resources may be inserted using the HTML-like hyperlink `<a>`. Its mandatory attribute `href` contains URL of the external resource.

The `<a>` element can have any inline content except the following elements: `<a>`, `<xref>` and `<cite>`. In XHTML output, the reference will be rendered as a standard hyperlink, whereas the typeset output will have the URL in a footnote.

## 7.2 Index Entries

Techrep2 also provides means for marking terms and phrases in the text as index entries. While this is not immediately useful for individual technical reports, an index may be desirable for proceedings and other publications.

An index entry is represented by the `<index>` element and its inline content. The following optional attributes control the formatting of index entries:

`silent`

This attribute with the value “true” indicates a *silent* index entry, which will appear in the index but not in the main text. The default is “false”.

`under`

This attribute indicates a second-level index entry, which will appear in the index under the first-level entry given as the value of this attribute.

`role`

This attribute specifies special rendering for the index entry. The choices are:

- `no`: no special rendering (default);
- `it`: italics;
- `tt`: monospaced font;
- `bn`: the *page number* of the index entry will be set in boldface;
- `un`: the *page number* of the index entry will be underlined;
- `xe`: XML element: the entry will be enclosed by chevrons “<” and “>”.

## 8 Bibliographic References

The list of bibliographic references is a special section enclosed in the `<biblist>` element. It is optional but if present, it must always appear after all `h1`, `h2` and `h3` sections and their text, and before the first `<appendix>`, if there is any.

The `<biblist>` element contains one or more `<bibitem>` elements. Each `<bibitem>` must have the `xml:id` attribute that defines the ID used in `<cite>` references.

The contents of `<bibitem>` elements are not further structured, any hybrid content is allowed.

## 9 Available Tools

XSLT stylesheets and other tools for processing technical reports are available on-line<sup>5</sup> under the GNU GPL license [6]. While the stylesheets and T<sub>E</sub>X formats are specifically tailored to the presentation style of CESNET technical reports, it should be easy to modify them for other purposes.

The distribution tarball comprises the following files:

*dbk2tr.xsl*

Translates DocBook v. 4.5 to Techrep2.

*hlasform.py*

Python script that generates the voting form for selecting report for proceedings from an Org file.

*hlaszprac.py*

Python script that processes the proceeding votes.

*license.txt*

Full text of the GNU GPL version 3 license.

*Makefile.odt*

Makefile for reports submitted as ODT.

*Makefile.rest*

Makefile for reports submitted as reStructuredText or DocBook.

*Makefile.tr1*

Makefile for reports submitted as Techrep (v1 or v2).

*odt2tr.xsl*

Translates Open Document Text to Techrep1.

*org2cesnet.xsl*

Adds references to CESNET CSS stylesheets to HTML generated by Emacs Org mode<sup>6</sup>.

*plasTeX/\_\_\_init\_\_\_.py*

Techrep v1 renderer for plasTeX (Python tool that parses L<sup>A</sup>T<sub>E</sub>X).

*rxml2tr.xsl*

Translates reStructuredText XML to Techrep2.

*Schema/techrep2.rng*

Annotated RELAX NG schema for Techrep2.

*Schema/Makefile*

Makefile that transforms the above schema to compact syntax and also generates pretty-printed HTML version.

<sup>5</sup> <http://staff.cesnet.cz/~lhotka/dist/techrep-current.tar.gz>

<sup>6</sup> <http://orgmode.org/>

*techrep.tex*

XeTeX format for individual technical reports.

*trcommon.tex*

Common TeX macros.

*trproc.tex*

XeTeX format for proceedings.

*trto-lib.xsl*

Common XSLT templates.

*trtors.xsl*

RSS news generator.

*trtotex.xsl*

Translates Techrep2 to T<sub>E</sub>X source (to be used with either techrep.tex or trproc.tex format).

*trtoxhtml.xsl*

Translates Techrep2 to XHTML.

*tr1to2.xsl*

Translates Techrep1 to Techrep2.

## 10 Acknowledgements

Pavel Kácha, Pavel Satrapa and Milan Sova significantly contributed to the discussions leading to the Techrep2 design and/or wrote XSLT stylesheets for Techrep1 that served as the basis for Techrep2 tools.

## References

- [1] BERNERS-LEE, T.; FIELDING, R.; MASINTER, L. *Uniform Resource Identifier (URI): Generic Syntax*. RFC 3986<sup>7</sup>, IETF, January 2005.
- [2] BRAY, T. et al. (ed.). *Namespaces in XML 1.0 (Second Edition)*. W3C Recommendation, 16 August 2006 [cit. 2009-12-01]. Available online<sup>8</sup>.
- [3] BRAY, T. et al. (ed.). *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C Recommendation, 26 November 2008 [cit. 2009-12-01]. Available online<sup>9</sup>.
- [4] DURUSAU, P.; BRAUER, M.; OPPERMAN, L. (ed.). *Open Document Format for Office Applications (OpenDocument) v1.1*. OASIS, 2007 [cit. 2009-12-01]. Available online<sup>10</sup>.

<sup>7</sup> <http://tools.ietf.org/html/rfc3986>

<sup>8</sup> <http://www.w3.org/TR/REC-xml-names/>

<sup>9</sup> <http://www.w3.org/TR/xml>

<sup>10</sup> <http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1-html/OpenDocument-v1.1.html>

- [5] GOODGER, D. *reStructuredText Markup Specification*. [cit. 2009-12-01]. Available online<sup>11</sup>.
- [6] *GNU General Public License*. Version 3. Free Software Foundation, 29 June 2007 [cit. 2010-02-15]. Available online<sup>12</sup>.
- [7] *Information Technology – Document Schema Definition Languages (DSDL) – Part 2: Regular-Grammar-Based Validation – RELAX NG*. Second Edition. ISO/IEC International Standard 19757-2:2008(E)<sup>13</sup>. December 2008.
- [8] LAMPORT, L. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*. Boston (MA): Addison-Wesley, 1994. xiii, 272 p. ISBN 978-0-201-52983-8.
- [9] MARSH, J.; VEILLARD, D.; WALSH, N. *xml:id Version 1.0*. W3C Recommendation, 9 September 2005 [cit. 2009-12-01]. Available online<sup>14</sup>.
- [10] PHILLIPS, A.; DAVIS, M. (ed.). *Tags for Identifying Languages*. RFC 5646<sup>15</sup>, IETF, September 2009.
- [11] SATRAPA, P. *Formát XML pro interní dokumentaci TEN-155 CZ*. [XML format for internal documentation of TEN-155 CZ project.] Technical report 1/2000<sup>16</sup>, Praha: CESNET, 2000.
- [12] WALSH, N.; MUELLNER, L. *DocBook: The Definitive Guide*. Sebastopol (CA): O’Reilly and Assoc., 1999. xiii, 635 p. ISBN 978-1-565-92580-9.

---

<sup>11</sup> <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html>

<sup>12</sup> <http://www.fsf.org/licensing/licenses/gpl.html>

<sup>13</sup> [http://standards.iso.org/ittf/PubliclyAvailableStandards/c052348\\_ISO\\_IEC\\_19757-2\\_2008\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c052348_ISO_IEC_19757-2_2008(E).zip)

<sup>14</sup> <http://www.w3.org/TR/xml-id/>

<sup>15</sup> <http://tools.ietf.org/html/rfc5646>

<sup>16</sup> <http://www.cesnet.cz/doc/techzpravy/2000-1>

## Appendix A. RELAX NG Schema for Techrep2

This appendix contains the complete schema for Techrep2 in the RELAX NG compact syntax. The annotated version of the same schema in the XML syntax is available online<sup>17</sup>.

```
#
# techrep2.rng - annotated RELAX NG schema for Techrep2
# Copyright © 2010 CESNET
# Author: Ladislav Lhotka <Lhotka@cesnet.cz>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/>.
#
default namespace = "http://cesnet.cz/ns/techrep/base/2.0"
namespace a = "http://cesnet.cz/ns/rngrest-annotations/1.0"
start = element report { report-content }
# Common patterns
common-attributes = lang-attribute?, role-attribute?
id-attribute = attribute xml:id { xsd:ID }
lang-attribute = attribute xml:lang { "en" | "cs" }
role-attribute = attribute role { xsd:token }
report-content =
  common-attributes,
  id-attribute?,
  attribute status { "proof" | "final" }?,
  attribute number { text }?,
  element title {
    common-attributes,
    attribute short { text }?,
    inline-content
  },
  element authors { authors-content },
  element date { date-content }?,
  (element keywords { keywords-content }
```

<sup>17</sup> <http://cesnet.cz/ns/techrep/base/2.0>

```

    & element abstract { hybrid-content }},
    element body { body-content }
date-content = common-attributes, text
authors-content =
    element affiliation { id-attribute, text }*
    & element author {
        attribute affil { xsd:IDREFS }?,
        attribute email { text }?,
        text
    }+
email-content = common-attributes, text
keywords-content = common-attributes, text
hybrid-content = p-content | (common-attributes, block-choice*)
inline-content =
    common-attributes,
    mixed { inline-choice* }
inline-choice =
    tt-element
    | sup-element
    | sub-element
    | em-element
    | strong-element
    | phrase-element
    | command-element
    | input-element
    | uri-element
    | file-element
    | footnote-element
    | q-element
    | a-element
    | xref-element
    | cite-element
    | index-element
    | br-element
block-choice =
    p-element
    | pre-element
    | blockquote-element
    | image-element
    | tabular-element
    | ol-element
    | ul-element
    | dl-element
toplevel-choice = block-choice | figure-element | table-element
body-content =
    common-attributes,

```

```

toplevel-choice*,
(element h1 { id-attribute?, inline-content },
sect1-content)*,
(element biblist { biblist-content }?
& (element appendix { id-attribute?, inline-content },
sect1-content)*)
sect1-content =
(toplevel-choice
| (element h2 { id-attribute?, inline-content },
sect2-content))+
sect2-content =
(toplevel-choice
| (element h3 { id-attribute?, inline-content },
toplevel-choice))+
biblist-content =
common-attributes,
element bibitem { bibitem-content }+
bibitem-content = id-attribute, common-attributes, inline-content
# Inline elements
tt-element = element tt { inline-content }
sup-element = element sup { inline-content }
sub-element = element sub { inline-content }
em-element = element em { inline-content }
strong-element = element strong { inline-content }
phrase-element =
common-attributes,
element phrase { inline-content }
command-element = element command { inline-content }
input-element = element input { inline-content }
uri-element = element uri { inline-content }
file-element = element file { inline-content }
footnote-element = element footnote { hybrid-content }
q-element = element q { inline-content }
a-element =
element a {
common-attributes,
attribute href { xsd:anyURI },
mixed { a-choice* }
}
a-choice =
tt-element
| sup-element
| sub-element
| em-element
| strong-element
| phrase-element

```

```

| command-element
| input-element
| uri-element
| file-element
| footnote-element
| q-element
| br-element
| index-element
xref-element =
  element xref {
    attribute raw { xsd:boolean }?,
    attribute linkend { xsd:IDREF },
    inline-content
  }
cite-element =
  element cite {
    common-attributes,
    attribute bibref { xsd:IDREF }
  }
index-element =
  element index {
    lang-attribute?,
    attribute silent { xsd:boolean }?,
    attribute under { text }?,
    attribute role { "no" | "it" | "tt" | "bn" | "un" | "xe" }?,
    inline-content
  }
br-element = element br { common-attributes, empty }
# Block elements
pre-element =
  element pre {
    common-attributes,
    attribute numbered { xsd:boolean }?,
    text
  }
blockquote-element =
  element blockquote { common-attributes, block-choice* }
p-element = element p { p-content }
p-content =
  common-attributes,
  mixed { p-choice* }
p-choice =
  tt-element
  | sup-element
  | sub-element
  | em-element

```

```

| strong-element
| phrase-element
| command-element
| input-element
| uri-element
| file-element
| footnote-element
| q-element
| a-element
| xref-element
| cite-element
| index-element
| br-element
| element ol { compact-li-element+ }
| element ul { compact-li-element+ }
image-element =
  element image {
    common-attributes,
    element source {
      common-attributes,
      attribute format { format-choice },
      attribute file { xsd:anyURI }
    }+
  }
format-choice = "EPS" | "GIF" | "JPEG" | "PDF" | "PNG" | "SVG"
tabular-element =
  element tabular {
    common-attributes,
    attribute colspec {
      xsd:token { pattern = "[lcr]+" }
    },
    tr-element+
  }
tr-element =
  element tr {
    common-attributes, bgcolor-attribute, (td-element | th-element)+
  }
bgcolor-attribute =
  attribute bgcolor {
    xsd:token { pattern = "#[0-9a-fA-F]{6}" }
  }?
td-element = element td { tabular-cell-content }
th-element = element th { tabular-cell-content }
tabular-cell-content =
  bgcolor-attribute,
  attribute align { "left" | "center" | "right" }?,

```

```

    attribute colspan { xsd:positiveInteger }?,
    hybrid-content
figure-element =
    element figure {
        id-attribute?, common-attributes, (block-choice+ & caption-element)
    }
table-element =
    element table {
        id-attribute?, common-attributes, (block-choice+ & caption-element)
    }
caption-element = element caption { hybrid-content }
ol-element =
    element ol {
        id-attribute?,
        attribute continue { xsd:IDREF }?,
        attribute labels { labels-choice }?,
        list-content
    }
labels-choice = "arabic" | "roman" | "ROMAN" | "alpha" | "ALPHA"
ul-element = element ul { list-content }
list-content =
    common-attributes,
    ((attribute compact { "true" },
        compact-li-element+)
    | (attribute compact { "false" }?,
        li-element+))
li-element = element li { id-attribute?, hybrid-content }
compact-li-element = element li { inline-content }
dl-element =
    element dl { common-attributes, (dt-element+, dd-element)+ }
dt-element = element dt { id-attribute?, inline-content }
dd-element = element dd { hybrid-content }

```

## Appendix B. Techrep2 Element Tags

This appendix contains an alphabetically sorted list of all XML element tags of the Techrep2 language together with short descriptions and links to the sections where the syntax and semantics of the particular element is explained.

Tag	Description	Section
a	hyperlink	Section 7.1
abstract	report abstract	Section 4.4
affiliation	author's affiliation	Section 4.2
appendix	appendix section	Section 5
author	author name	Section 4.2
authors	container for authors	Section 4.2
bibitem	bibliography item	Section 8
biblist	bibliographic list	Section 8
blockquote	block of quoted text	Section 6.2
body	report body	Section 2
br	line break	Section 7
caption	figure or table caption	Section 6.5
cite	bibliographic reference	Section 7.1
command	operating system command	Section 7
date	report submission date	Section 4.3
dd	definition description	Section 6.4
dl	definition list	Section 6.4
dt	definition term	Section 6.4
em	text emphasis	Section 7
figure	floating figure	Section 6.5
file	file name	Section 7
footnote	footnote	Section 7.1
h1	top-level section	Section 5
h2	second-level section	Section 5
h3	third-level section	Section 5
image	graphic image	Section 6.6
index	index entry	Section 7.2
input	user-entered input	Section 7
keywords	report keywords	Section 4.5
li	list item in<ul> and<ol> lists	Section 6.4
ol	ordered list	Section 6.4
p	text paragraph	Section 6.1
phrase	user-defined text phrase	Section 7
pre	preformatted text	Section 6.3
q	text in double quotes	Section 7
report	report (document element)	Section 2
source	source file for an image	Section 6.6
strong	strong text emphasis	Section 7
sub	subscript	Section 7

sup	superscript	Section 7
table	floating table	Section 6.5
tabular	tabular object	Section 6.7
td	table data	Section 6.7
th	header cell in a table	Section 6.7
title	report title	Section 4.1
tr	table row	Section 6.7
tt	text in a monospaced font	Section 7
ul	unordered list	Section 6.4
uri	uniform resource identifier	Section 7
xref	cross reference	Section 7.1