Frequently Asked Questions about GELLMU

William F. Hammond

Last Revision: July 5, 2007

Table of Contents

Part I Basic Questions . . . 1

1 What is GELLMU? . . . 1

2 What is XML? . . . 2

3 Why was GELLMU Created? . . . 2

4 Can GELLMU Help Me to Write My Web Pages? . . . 2

5 What is the Diffence between Basic and Regular GELLMU? . . . 2

6 Are There Other Similar Markups? . . . 3

Part II Running GELLMU . . . 3

7 On What Platforms Can GELLMU Be Used? . . . 3

8 What is Needed to Run GELLMU? . . . 3

Part III Things in LaTeX not in GELLMU . . . 4

9 Why are So Many Things Not “In” GELLMU? . . . 4

10 Why Doesn’t `\ref` Automatically Make Anchors? . . . 4

11 What About Split-File Source? . . . 4

12 What About Split-File Formattings? . . . 5

Part IV Relation of GELLMU to Other Things . . . 5

13 How is GELLMU Related to *latex2html*? . . . 5

14 How is GELLMU Related to *hyperlatex*? . . . 5

15 How is GELLMU Related to *tex4ht*? . . . 6

Part I Basic Questions

1 What is GELLMU?

GELLMU stands for “Generalized Extensible LaTeX-Like MarkUp”. It is a markup language with syntax resembling in large part that of LaTeX that is designed to provide a bridge for authors who wish to migrate from LaTeX to XML (see question 2).

2 What is XML?

XML stands for “Extensible Markup Language”, which is second generation basic markup language for the World Wide Web. The use of the word *extensible* means that the set of markup tags is not fixed. That, in turn, means that XML is not a markup language but rather a family of markup languages with a shared syntax.

3 Why was GELLMU Created?

The author is a math professor who became frustrated in recent years with the extra work required to make his course materials available on the web for his students.

He found that if he wrote HTML directly, he could not derive adequate printed versions, and he found converters for translating LaTeX into HTML cumbersome, inaccurate, and unsatisfactory[1].

GELLMU is designed as a single source language in which to write from which different formats can be automatically derived with complete reliability. For more on single source, multiple output languages for authors see question 6.

4 Can GELLMU Help Me to Write My Web Pages?

Yes, in two ways. There is a choice between *basic* GELLMU and *regular* GELLMU.

If you are familiar with the tag set for HTML (the basic language used in web pages) and do not want any form for what you write other than a web page, then *basic* GELLMU provides LaTeX-like notation, which is somewhat more succinct than regular HTML tagging, and *newcommand*-like macros for writing HTML. For this see also part IV.

If you want to write for more than one ultimate output with a single source markup (see question 6), then *regular* GELLMU may be considered.

5 What is the Diffence between Basic and Regular GELLMU?

Basic GELLMU provides minimal (1) LaTeX-like syntax and (2) *newcommand*-like macros for conscious writing toward an SGML or XML document type such as HTML, DocBook, TEI, SGMLTexi, ...

Regular GELLMU provides also a fuller array of LaTeX-like syntax features and some LaTeX-like shorthand. Moreover, the GELLMU didactic production system provides a document type with many LaTeX-like commands.

[1] There are now better converters; however, the overall idea of translating from LaTeX to HTML is not logically sound. (On the other hand, it *is* logically sound to write a translator from HTML to LaTeX.)

6 Are There Other Choices for Single Source, Multiple Output Markup Languages for Authors?

Yes. *Texinfo*, the language of the GNU[2] Documentation System, is an example. In the 80's *Texinfo* could be used to provide two forms of a GNU document: (1) *Info* online hypertext, suitable for the text-based terminals with cursor control of that time and (2) finely typeset text in DVI format.

By the late 80's Charles Goldfarb had created SGML, the very general family of markup languages in which HTML is a member and relative to which XML is a subfamily. Although it is formally possible for a language under this umbrella to admit essentially no translation, the intention was to provide powerful languages suitable for multiple outputs and to provide, thanks to the shared syntax, general processing frameworks under which translations can be efficiently and reliably written.

Part II Running GELLMU

7 On What Platforms Can GELLMU Be Used?

A complete suite of materials for the GELLMU didactic production system is available for the GNU Operating System, Linux, the various flavors of Unix, and modern "Windows".

8 What is Needed to Run GELLMU?

Minimally the GELLMU Syntactic Translator, an GNU Emacs Lisp program. To use it, you need the Emacs Lisp engine, which is GNU Emacs.

For *basic* GELLMU (see question 5) the GELLMU Syntactic Translator is all that you need. Its output is the SGML or XML language such as, for example, HTML, for which you are explicitly tagging. Of course, you need whatever is required in order for you to deal with that output.

For *regular* GELLMU with the didactic GELLMU production system you need to have

1. James Clark's *nsgmls/SP*.
2. Perl.
3. TEX.

[2] The free software project of Richard Stallman and the Free Software Foundation. See `http://www.gnu.org/`.

Part III Things in LaTeX not in GELLMU

9 Why are So Many Things Not "In" GELLMU?

The didactic document type is intended to be a base for further development. For that reason its design is minimalist. Its author believes that the techniques used are adequate for a substantially more elaborate system. One who so wishes can move it closer to regular LaTeX than it already is. On the other hand, if it is moved too close to LaTeX, which is a structured typesetting language, then the range of possible translation targets might be diminished.

More information about this will be found in the GELLMU Manual[3], especially in the appendix entitled *Release Notes*[4].

10 Why Doesn't `\ref` Automatically Make Anchors?

First note that every `\label` command does result in the creation of a base to which one may anchor. In the two didactic formattings for LaTeX and HTML the anchor base is not visible when rendered[5]. There are three reasons why `\ref` does not automatically provide a link:

1. Since *ref* is the same as a name in LaTeX, there is an effort to emulate LaTeX's behavior in this regard.
2. There is more than one reasonable way to "style" an anchor.
3. It is extremely easy for an author to use *newcommand* to reference a label. For example, if an author writes

   ```
   \newcommand{\Ref}[1]{\anch[iref="\ref{#1}"]{\ref{#1}}} ,
   ```

   then *Ref* is created as a name for providing a label reference (by key, as with *ref*) with an anchor. Or the same newcommand with name string `"ref"` instead of `"Ref"` is also allowed in GELLMU, though in LaTeX one would need to use *renewcommand*[6].

11 What About Split-File Source?

The best way to merge document input is at the level of SGML source, i.e., at the stage represented by the output of the syntactic translator.

For example, this will make the parsing error reports from *nsgmls* easy to locate in GELLMU source since errors are reported by file and line number. (The syntactic translator preserves line number alignment between its source and its output, file by file.) Also when the syntactic translator is called separately for different parts of a document, the *newcommand* definitions are localized to each part, thereby making the resolution of name conflicts easy to track.

[3] URI: ../doc/glman.html

[4] URI: ../doc/glman.html#release

[5] Some HTML renderers might disclose *id* anchors with empty content.

[6] Since LaTeX's *newcommand* actually creates a new command whereas GELLMU's *newcommand* is one-time forward macro substitution.

12 What About Split-File Formattings?

As it is, the didactic production system is based on one file per document. If that is not what you want, just customize your formatters to produce what you want.

This would make sense, for example, for a *collection* document class that was constructed to consist of independently authored GELLMU *article*s. While one might want a unified typesetting, one might also want separate HTML versions and separate screen-sized PDF versions.

In a design like that of the didactic production system the document is seen as a whole at the stage where cross-referencing is handled. This makes it possible for cross-referencing in split-file formattings to comprehend the parts as they relate to the whole.

Part IV Relation of GELLMU to Other Things

13 How is GELLMU Related to latex2html?

GELLMU provides LaTeX-like markup for the conscious writing of instances under an XML document type. That document type could be HTML.

latex2html, on the other hand, is designed to translate source consciously written as classical TeX or LaTeX markup.

14 How is GELLMU Related to hyperlatex?

GELLMU provides a way to write HTML consciously using LaTeX-like syntax, while *hyperlatex* involves translation of a special dialect of LaTeX. There are a number of similar features in *hyperlatex* including:

1. The use of LaTeX (not LaTeX-like) markup to write HTML.
2. The use of a GNU Emacs Lisp program to translate source markup (in this case to HTML).
3. Careful, reliable production of both HTML and typeset forms of a document.

As it stands in relation to SGML or XML, *hyperlatex* is specific for HTML. The source of a *hyperlatex* document may be given to *hyperlatex*'s Emacs Lisp program for translation to HTML or to *LaTeX*, the program, for typesetting with *hyperlatex* macros.

GELLMU's basic layer, which operates at the level of syntax without knowledge of markup vocabulary, may be used for writing under many SGML or XML document types, not just HTML. Subsequent to syntactic translation from LaTeX-like markup to standard SGML or XML markup, one may employ any suitable SGML or XML processor for further translation, typesetting, or other processing. GELLMU's basic layer for writing in HTML (or in any other SGML or XML document type) directly does not provide typesetting.

The GELLMU didactic production system provides what is intended as a first SGML/XML document type for those with experience as LaTeX authors. The system is designed for free-standing separate components, and new components can be introduced for translating to new formats. The use of *LaTeX*, the Program, with GELLMU source is a choice, not a requirement.

15 How is GELLMU Related to tex4ht?

The *tex4ht* family of applications is conversion software. It uses *TeX*, the Program, configured with special macros aimed at a target format, such as HTML, to create DVI "specials" in *TeX*'s normal DVI output. While one can print the DVI file in the usual way, one may also then submit it to a special post-processor for writing a translation of the original document in the target format. One can achieve very good results this way **if** the original source is carefully crafted. For legacy source this method is likely to produce optimal results.

GELLMU provides LaTeX-like markup for the conscious writing of instances under an XML document type. There is very little intersection of this with the functionality of *tex4ht* except in the relatively rare instances of TeX or LaTeX source that is carefully crafted under a custom markup.

GELLMU does not provide a technique for automatic translation of legacy documents although one may sometimes, especially with clever use of its macro facilities, succeed with human effort to "port" legacy LaTeX source into useful GELLMU source. The item `lfaq.glm` in the examples archive[7] is an example of a port.

[7] URI: ./