CV formatting with CurVe

Didier Verna

Abstract

CurVe is a LaTeX 2ε class package for writing curricula vitæ. It provides a set of commands to create headers, rubrics, entries in these rubrics and so on. CurVe will then format your CV with a consistent layout, while you can just concentrate on the contents. CurVe has a very special feature known as the *flavor mechanism*: it is able to manage different "flavors" (versions) of your CV simultaneously.

CurVe is distributed under the terms of the LPPL license. This paper gives an overview of the features available in version 1.4.

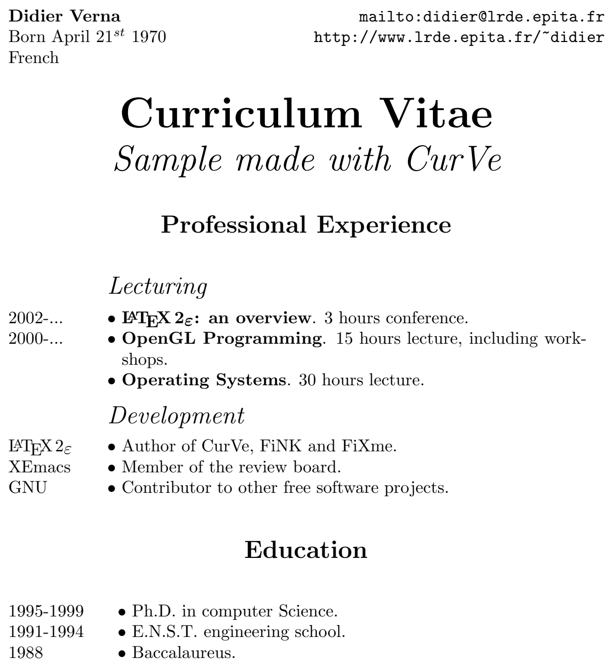
— — ∗ — —

1 Yet another CV class?

The first draft of CurVe appeared in 2000, when I was completing my Ph.D. and was starting to look for a job. At that time, several options were available: using a WYSIWYG editor, using plain (LA)TeX (that is, with no particular class), or using an already existing CV class for LaTeX. After some investigation, none of these options turned out to be satisfactory for me.

As the visual aspect of a CV is something very important, one could be tempted to use a WYSIWYG editor, thinking that the layout conception would be faster. For me, this was (and still is) wrong:

- Firstly, WYSIWYG editors are usually far from it: what you see (on screen) is most of the time *not* what you get (on paper). Hence, WYSIWYG editors often turn out to be a handicap rather than a help.
- Secondly, a good CV requires a very strict layout, of the kind on which you spend hours, unless it is completely automated. Doing this with a WYSIWYG editor would involve a deep knowledge of the editor itself (for instance, knowing



Didier Verna mailto:didier@lrde.epita.fr
Born April 21st 1970 http://www.lrde.epita.fr/~didier
French

Curriculum Vitae
Sample made with CurVe

Professional Experience

Lecturing

2002-...	• **LaTeX 2ε: an overview**. 3 hours conference.
2000-...	• **OpenGL Programming**. 15 hours lecture, including workshops.
	• **Operating Systems**. 30 hours lecture.

Development

LaTeX 2ε	• Author of CurVe, FiNK and FiXme.
XEmacs	• Member of the review board.
GNU	• Contributor to other free software projects.

Education

1995-1999	• Ph.D. in computer Science.
1991-1994	• E.N.S.T. engineering school.
1988	• Baccalaureus.

Figure 1: CurVe sample output

how to use *styles* instead of *manually* formatting all paragraphs the same way). Hence, using a WYSIWYG editor *properly* turns out to be very close to using LaTeX itself. Since I already knew LaTeX, it would have been a waste of time to learn yet another tool.

For an interesting discussion on WYSIWYG solutions for LaTeX, see Kastrup (2002).

The next solution was to use an already existing class or style with LaTeX. For reasons that I won't mention here, but that can be found in its documentation's introduction, the only reasonable option was `currvita` (Reichert, 1999). This solution did not satisfy me, mainly because the layout was too limited for me. Layout is probably something very personal in this context, but I also believe that there are different traditions in different countries, and that CurVe provides a layout that pleases many people, notably in France. The increasing number of people using CurVe tends to show that I am not mistaken.

So I decided to use LaTeX from scratch (some other reasons for this are explained later). Being a lazy kind of person however, I tried *not* to reinvent the wheel, and I found that David Carlisle's `LTXtable` package would do most of the formatting for me. Being also a free software kind of person, I wanted other people to benefit from my work, so I naturally made it a proper class with complete documentation.

The first release of CurVe occurred in February 2001. Today, CurVe is a simple yet powerful class, consisting of only 350 lines of code, probably half of which is devoted to handling user customizations.

2 Layout

The primary purpose of CurVe is to offer a set of predefined commands to specify the contents of your CV, while removing from you the burden of formatting it. This has two important consequences, however: CurVe requires that you conform to its document structuring scheme, and will expect that you like the way it formats the output.

Figure 1 shows the output of CurVe on a very small sample CV. There is no user-level customization in this example.

2.1 Headers

As you can see in the example, a CurVe CV begins with two optional headers (upper left and upper right) in which you usually put your name, address, email, whether you're married and so on. These headers will respectively be left and right aligned. CurVe also lets you insert a small identity photo in the headers, either on the left, on the right, or between them. After these headers comes an optional title and an optional subtitle, which will be centered on the page.

The headers and titles in this example are specified in the document's preamble as follows:

```
\leftheader{%
  \textbf{Didier Verna}\\
  Born April 21$^{st}$ 1970\\
  French}

\rightheader{%
  \url{mailto:didier@lrde.epita.fr}
  \url{http://www.lrde.epita.fr/~didier}}

\title{Curriculum Vit\ae}
\subtitle{Sample made with CurVe}
```

Later on, after the call to `\begin{document}`, these headers are formatted like this:

```
\makeheaders[t]
\maketitle
```

This scheme is very traditional in LaTeX classes and should not surprise anybody. The optional argument to `\makeheaders` specifies the vertical alignment. Here, **t**op is used. Many more commands are available to customize the appearance of the headers (positioning, size, fonts, etc.).

2.1.1 Rubrics

The remainder of the document is composed of sections called "rubrics" in the CurVe terminology. A rubric represents a major topic that you want to detail in your CV. Typical rubrics, as demonstrated in the example, are "Education", "Professional Experience" and the like. Rubrics have a title (which will be centered) and appear under the form of properly aligned "entries" (see below). If a rubric has to be split across different pages, its title will be repeated automatically.

Here is the code used to generate the "Professional Education" rubric:

```
\begin{rubric}{Professional Experience}
\subrubric{Lecturing}
\entry*[2002-...] \textbf{\LaTeXe: an
    overview}. 3 hours conference.
\entry*[2000-...] \textbf{OpenGL Programming}.
  15 hours lecture, including workshops.
\entry* \textbf{Operating Systems}.
  30 hours lecture.

\subrubric{Development}
\entry*[\LaTeXe] Author of CurVe, FiNK and
  FiXme.
\entry*[XEmacs] Member of the review board.
\entry*[GNU] Contributor to other free
  software projects.
\end{rubric}
```

As you can see, each rubric is made within a `rubric` environment, which takes the rubric's title as a mandatory argument. The other commands will be explained below. Please note that for technical reasons due to the use of the `LTXtable` package, each rubric must be written in a separate file. This is not a heavy burden however, and it even made things easier when I implemented the "flavor" mechanism described in the next section.

2.1.2 Subrubrics

You might want to further split your rubrics into different "subrubrics". For instance, figure 1 shows two subrubrics named "Lecturing" and "Development" under the "Professional Experience" rubric. Subrubrics' names are displayed in alignment with the entries' contents (see below), but are formatted differently so that they remain distinguishable. Subrubrics are created with the `\subrubric` command. For instance:

```
\subrubric{Development}
```

2.1.3 Entries

An entry is a final item of information related to the (sub)rubric under which it appears. An entry has a "contents", and an optional "key" under which it is classified. For instance, consider the "XEmacs" entry in the example:

```
\entry*[XEmacs] Member of the review board.
```

The key is "XEmacs" and the entry's contents is "Member of the review board.". Keys are usually used to indicate dates. CurVe aligns both keys and contents together. Keys are optional (hence, they should be input within brackets) in order for you to classify several entries together (without repeating the same key over and over again). For instance:

```
\entry*[2000-...] \textbf{OpenGL Programming}.
  15 hours lecture, including workshops.
\entry* \textbf{Operating Systems}.
  30 hours lecture.
```

Here, the second key is not repeated because the "Operating Systems" course began in the same year as the OpenGL one.

You probably have noticed already that a "starified" version of the `\entry` command is used. Actually, `\entry*` commands in `rubric` environments are very similar to `\item` commands in `list` environments. The reason for this "star-ification" is that the first version of CurVe had a somewhat ill-designed, unstarred `\entry` command that took the whole entry's contents as a second argument. The new scheme is much more elegant, but backward compatibility has been preserved.

3 The "flavor" mechanism

This is a very nice feature of CurVe, and one of the reasons that made me write it in the first place.

It is often desirable to maintain several slightly divergent versions of one's CV at the same time. For instance, when I was looking for a job back in 2001, I had a version of my CV emphasizing Artificial Intelligence, and another emphasizing Distributed Virtual Reality. Only the title and some entries in the "Professional Experience" rubric were a bit different; all other parts basically remained the same. CurVe provides an easy-to-use mechanism for maintaining different "flavors" of your CV at the same time. You basically write different versions of (some of) your rubrics in different files (this is needed for technical reasons anyway, as mentioned previously), tell CurVe which flavor you want to format (CurVe can even ask you which one to use directly). CurVe will then use the global skeleton, and whenever it finds a rubric file specialized for that particular flavor, it will use it. Otherwise, it will simply fall back to the default one (no particular flavor).

With a clever use of Makefiles, you can even manage to compile different flavors at the same time. Suppose you have a mainstream CV (let's call it `cv.tex`), with an "education" rubric in a file named `education.tex` and a "skills" one in `skills.tex`. Suppose further that you want a slightly divergent version of your CV with an alternate skills rubric.

The normal way to compile the alternate version is to save a new rubric file called, for instance, `skills.alt.tex`, and tell CurVe to use it by calling `\flavor{alt}` in your document's preamble. The only problem is that this alternate version will also compile to `cv.dvi` because both versions share the same skeleton and other rubric files.

To remedy this problem, you can make CurVe ask you at run-time which flavor to compile (this is done with the `ask` class option) and use special Makefile rules to answer the question for you, build the different flavors and move the different output files to flavor-specific names. Here is a Makefile example that would allow you to do this for as many different versions of `skills.tex` as you wish:

```
FLAVORS = alt # make other flavors if you want

all: all_flavors cv.dvi

all_flavors:
        for i in $(FLAVORS) ; do         \
          $(MAKE) cv.$$i.dvi FLAVOR=$$i ; \
        done

cv.$(FLAVOR).dvi: cv.tex education.tex \
                  skills.$(FLAVOR).tex
        echo $(FLAVOR) | latex cv.tex
        mv cv.dvi $@

cv.dvi: cv.tex education.tex skills.tex
        echo | latex cv.dvi
```

As you can see, the trick is to fork a new `make` subprocess for each flavor you want to build, and obviously finish with the mainstream version. An `echo` shell command is used in conjunction with the `ask` class option to indicate the current flavor to build. For each `make` subprocess, this flavor is stored in the dynamic variable `FLAVOR`, and the corresponding `cv.$(FLAVOR).dvi` rule is used, with the proper dependencies.

An implementation sidenote. In order to implement the flavor mechanism, the LaTeX macro `\input` has been redefined to look for flavored files first. This is actually very nice because you can use it to make different flavors of text that do not belong in rubrics. For instance, suppose you want a special version of the subtitle of your CV for the flavor `alt`. Put your alternate subtitle in a file called `subtitle.alt.tex`; do something similar for the default subtitle. Now go to the skeleton of your CV, and write `\input{subtitle}` in the preamble. That's it. You'll have different subtitles in your different CV flavors.

4 Conclusion

In this paper, I preferred to concentrate on special aspects of CurVe's history or peculiarities rather than on its user interface, because the latter is relatively straightforward and fully described in the package documentation. Although the layout of CurVe is strict, it remains deeply customizable. CurVe also has other features that are worth mentioning, like support for all standard class options, support for the `.ltx` file extension if, like me, you prefer it over `.tex` for LaTeX files, support for standard LaTeX bibliographic commands (although I don't recommend using them), and support for AUC-TeX. Additionally, CurVe has been translated into six different languages (English, French, Spanish, German, Italian and Danish).

For an example of what you can do with CurVe, get my own CV at `http://www.lrde.epita.fr/~didier/perso/cv.php`. You will also find a page describing some formatting tricks I've used to make it more eye-catching.

Thanks to different contributors of code or suggestions, CurVe has evolved since its first release, and I hope it will continue to evolve and reach more and more users in the future.

References

Kastrup, David. "Revisiting WYSIWYG paradigms for authoring LaTeX". 2002. `http://preview-latex.sourceforge.net/wysiwyg-draft.pdf`.

Reichert, Axel. "`currvita.sty`". 1999. A curriculum vitae style for LaTeX, available in macros/latex/contrib on CTAN.

◇ Didier Verna
EPITA, Research and Development Laboratory,
14-16 rue Voltaire
94276 Le Kremlin-Bicêtre,
France
`didier@lrde.epita.fr`
`http://www.lrde.epita.fr/~didier`