LaTeX classes for doctoral theses in Ukraine: Interesting tips and painful problems

Oleksandr Baranovskyi

Abstract

In this paper, I introduce vakthesis, a bundle of LaTeX classes for typesetting doctoral theses according to official requirements in Ukraine, discuss the current status of the project and future development plans. Some LaTeX programming tricks that I have studied are considered.

1 Introduction

vakthesis is a bundle of LaTeX classes for typesetting doctoral theses (or dissertations) in Ukraine [7].

The bundle consists of the following main components: vakthesis and vakaref are traditional classes for a thesis and for a summary (автореферат in Ukrainian) respectively; while mon2017dev and mon-2017dev-aref are modern classes for a thesis and for a summary respectively.

Traditional classes conform to the now "obsolete" official format required by the VAK (Вища атестаційна комісія, i.e., Higher Attestation Commission). Now they are suitable for previously defended theses.

In 2017, the MON (Міністерство освіти і науки, i.e., Ministry of Education and Science) published its own style guide. Modern classes conform to this "new" official format by the MON. These classes are recommended for persons defending their theses now. They are based on the vakthesis and vakaref classes respectively and cannot work without them.

The vakthesis bundle contains a number of example files: the main LaTeX file of a thesis/summary, introduction, chapter 1, conclusion, bibliography, and BibTeX files.

So, a thesis author can use these example files as a template for his/her thesis and will receive a thesis/summary with a properly formatted title page of a thesis (and cover of a summary), headings of chapters, sections, subsections, etc., numeration for pages, chapters, sections, subsections, etc., as well as appendices, captions of figures and tables, theorems, lemmata, definitions, etc., list of references, and (in modern classes only) an abstract and list of keywords.

2 Tips and problems

In this section, I will discuss some programming tricks related to vakthesis development. The first and second cases give tips on how one can define a command with optional arguments of a special form and avoid BibTeX restrictions, respectively. The remaining three cases are devoted to the problems with UTF-8 encoding, checking of package loading, and building classes on the base of other classes.

2.1 Command `\speciality` with two optional arguments

In Ukraine, any thesis should be classified according to the so-called List of Specialities.

This is a list of correspondences between speciality code and speciality name and field of science. For example, 01.01.01 is a code for speciality математичний аналіз (Mathematical Analysis) and field of science фізико-математичні науки (Physical and Mathematical Sciences). In this case, a PhD student will be awarded the academic degree of кандидат фізико-математичних наук (Candidate of Physical and Mathematical Sciences).

In the vakthesis bundle, there is a particular command `\speciality`. In the standard situation it requires one mandatory argument:

```
\speciality{01.01.01}
```

For a given speciality code, this command takes a speciality name and a field of science from a special plain-text database. Then this information is typeset on the title page of the thesis.

The `\speciality` command has two optional arguments if, for some reason, the user needs to provide a speciality name or a field of science:

```
\speciality[математичний аналіз]{01.01.01}
  [фізико-математичних наук]
```

For example, some specialities in the list have an ambiguous form such as теорія та методика навчання (з галузей знань). It means Theory and Methodology of Learning (on some field of knowledge).

In this case the thesis author should choose an appropriate field of knowledge and write, for example, теорія та методика навчання математики (Theory and Methodology of Learning of Mathematics) on the title page.

On the other hand, for some specialities, a degree can be awarded on various fields of science. For example, for the speciality with code 01.04.07, a PhD student can obtain a degree either of кандидат фізико-математичних наук (Candidate of Physical and Mathematical Sciences) or of кандидат технічних наук (Candidate of Technical Sciences) according to the specific nature of the research.

These cases cannot be processed by an algorithm, so optional arguments of the `\speciality` command can be used:

```
\speciality[теорія та методика навчання
  математики]{13.00.02}
```

and

doi.org/10.47397/tb/43-2/tb134baranovskyi-vakthesis

```
\speciality{01.04.07}[технічних наук]
```

respectively. Of course both optional arguments may be used simultaneously as in the above-mentioned example for speciality code 01.01.01.

Hence the command `\speciality` cannot have the following syntax:

`\speciality[`⟨*something*⟩`][`⟨*something*⟩`]{`⟨*something*⟩`}`

and cannot be defined by the standard LaTeX command `\newcommand`.

To define the command with two optional arguments around the mandatory argument we can use commands `\def` and `\@ifnextchar`. Below I will give a simplified pseudocode to demonstrate the main idea.

```
\def\speciality{%
  \@ifnextchar[\a@cmd\aa@cmd
}
\def\a@cmd[#1]{%
  % Process speciality name #1
  \@speciality
}
\def\@speciality#1{%
  % Process speciality code #1
  \@ifnextchar[\a@cmd@b\a@cmd@bb
}
\def\a@cmd@b[#1]{%
  % Process field of science #1
}
\def\a@cmd@bb{%
  % Find field of science in the DB
}
\def\aa@cmd#1{%
  % Process speciality code #1
  \@ifnextchar[\aa@cmd@b\aa@cmd@bb
}
\def\aa@cmd@b[#1]{%
  % Process field of science #1
  % Find speciality name in the DB
}
\def\aa@cmd@bb{%
  % Find speciality name in the DB
  % Find field of science in the DB
}
```

This is a very simple idea; and this idea just works. But this code is difficult to maintain as well as to extend to three or more optional arguments.

This is a reason why I reject this idea in a modern version of the classes. Now I use the xkeyval package [1] to provide a simple key–value interface:

```
\speciality[
  specialityname=Теорія і методика професійної
    освіти,
  degreefield=педагогічні,
  % specialityfile=⟨filename⟩.csv
]{13.00.04}
```

A useful overview of how the LaTeX key–value system works is given in [8].

2.2 Environment `bibset` to make two reference lists in a thesis

It is known that BibTeX can generate only one reference list in a document, i.e., only one command `\bibliography` can be processed in standard situations. Nevertheless, sometimes there is a need to have more than one reference list. In particular, in Ukrainian theses, the following two lists may exist: the list of referenced sources and the list of author's publications.

There are many solutions for multiple bibliographies [5, 6] but they are not suitable for me.

Essentially, the multibbl and multibib packages provide a special bibliography "tag". So, special "tagged" `\cite` and `\bibliography` commands are available for the user.

Similarly, with the splitbib package, the user needs to categorize citations in the document.

The bibtopic package separates different bibliographies on different `.bib` files and uses special commands instead of the standard `\bibliography`.

The chapterbib and bibunits packages separate bibliographies per chapter or per other logical units.

In the vakthesis bundle, the `bibset` environment is provided that is used in the following way. Let `xampl-thesis.tex` be a thesis file containing the commands for the list of referenced sources and the list of author's publications.

```
\begin{bibset}{Список використаних джерел}
 \bibliographystyle{⟨bibliography style 1⟩}
 \bibliography{⟨referenced sources⟩}
\end{bibset}

\begin{bibset}[a]{Список публікацій автора}
 \bibliographystyle{⟨bibliography style 2⟩}
 \bibliography{⟨author's publications⟩}
\end{bibset}
```

Standard `\cite` and `\bibliography` commands are used in the text. There is no need to tag or categorize them. Argument of any `\bibliography` command may be any list of `.bib` files.

The `bibset` environment redefines commands `\bibliographystyle` and `\bibliography` such that they write commands `\bibstyle` and `\bibdata` to the `.aux` file if they appear in the first environment `bibset` and do not write otherwise.

Then during the first run BibTeX sees only one copy of `\bibstyle` and `\bibdata`, corresponding to the first environment `bibset`. At this point the file `xampl-thesis.bbl` generated by BibTeX should

be renamed to `xampl-thesis1.bbl` (manually or programmatically).

During the second run the first and second `bibset` environments are interchanged, i.e., commands `\bibstyle` and `\bibdata` are written to the `.aux` file if they appear in the second `bibset` environment and are not written otherwise.

So now BibTeX sees commands `\bibstyle` and `\bibdata` corresponding to the second `bibset` environment. At this point the file `xampl-thesis.bbl` should be renamed to `xampl-thesis2.bbl`.

At the last step LaTeX includes the generated files `xampl-thesis1.bbl` and `xampl-thesis2.bbl` at the corresponding places. Hence the author has two reference lists in the thesis.

In earlier versions of the official style guides by VAK there was not any mention of the two reference lists. I just realised this for myself. However, the modern official style guide by MON contains an explicit recommendation to prepare up to three reference lists in a thesis. This solution works in modern classes too.

2.3 The casus package and UTF-8

In the summary, authors write the name of the institution where they studied and worked on their thesis. So a command `\institution` is provided by classes whose argument is the name of this institution. This name (in the nominative case) appears on the cover page, for example, Національний педагогічний університет імені М. П. Драгоманова.

At the same time, there is a sentence on the reverse side of the cover page, where the author states that this work is performed at this institution. Here the name of institution is in the locative case, for example, у Національному педагогічному університеті імені М. П. Драгоманова.

In the Ukrainian language, a noun (as well as some other parts of speech) can change its form to express its syntactic function in the sentence. There exist seven cases of a noun: nominative, genitive, dative, accusative, instrumental, locative, and vocative.

For example, університет is a university in Ukrainian. This is the form of nominative case. If I want to write that I study at a university, I would use the form of locative case: я навчаюся в університеті. That is, the ending is changing and there is also a preposition.

I think it is redundant to ask a thesis author to provide different forms of the institution name if the `\institution` command already gives a nominative form of the name. The vakthesis or vakaref class can "compute" genitive, dative or other form. This is exactly what the auxiliary package casus does.

Briefly, the algorithm is the following. Suppose the institution name is Національний педагогічний університет імені М. П. Драгоманова. Here there is a special word університет from the list of "known words". All words before the known word are adjectives. The casus package has rules to decline adjectives as well as rules to decline nouns. All words after the known word should not be changed.

The algorithm splits a given sentence into words, finds a known word (університет, інститут, академія, коледж, міністерство, бібліотека, кафедра, etc.), then declines adjectives and the known word (as a noun), and then stops, i.e., do not touch the words after the known word. In Ukrainian, any name of institution has this form. So this algorithm works.

Unfortunately, one day an author reported to me a problem he encountered in his thesis. He just re-encoded all vakthesis files and his thesis files from Windows-1251 encoding to UTF-8 encoding.

After this operation he received some mysterious error messages such as

```
Missing number, treated as zero.
```

or

```
Undefined control sequence.
```

Skipping non-essential details, the main problem is in the casus package.

To decline a word (in particular, an adjective педагогічний) the algorithm runs through the word until it finds an ending from a given list of endings. This is the ending -ий for this word. This ending is skipped and the other ending that corresponds to a given case is added. So the words педагогічного, педагогічному and so on are received.

This simple idea should not depend on a file encoding. However, my implementation of the algorithm does not work if a file has an UTF-8 encoding. My conjecture is that the algorithm implementation fails for characters encoded by two or more bytes.

Maybe the better solution would be to use a known stable package for string manipulation instead of my quick and dirty solution implemented in the casus package.

2.4 Incorrect checking if hyperref is loaded

Traditionally, theses are printed on paper and stored at physical libraries. However, LaTeX can generate an electronic document with hyperlinks. In particular, the hyperref package can be used to this end. Some authors of theses want to use this possibility.

Generally speaking, it is complicated to use vakthesis classes and the hyperref package simultaneously. This can cause errors that are hard to diagnose.

In particular, vakthesis classes redefine some internal commands such as `\@spart`, `\@schapter`, and `\@ssect`. The number of arguments is even changed. The hyperref package makes its modifications carefully but cannot predict that these commands have more arguments now. As a result, sectioning commands do not work as expected.

Hence, to carefully interact with the hyperref package, vakthesis classes should check if hyperref is loaded, for example, by means of the internal command `\@ifpackageloaded`:

```
\@ifpackageloaded{hyperref}
  {⟨true branch⟩}
  {⟨false branch⟩}
```

In the ⟨*true branch*⟩, classes should define versions of commands and environments that are aware of the hyperref package.

In particular, this approach would allow solving the above-mentioned problem with sectioning commands.

But the internal command `\@ifpackageloaded` can be used in a document preamble only. I cannot use this command, for example, inside of the `bibset` environment to check if hyperref is loaded. So I check if some internal command of hyperref is defined:

```
\@ifundefined{hyper@warn}
  {⟨true branch⟩}
  {⟨false branch⟩}
```

It is easy; and it works. But, after a few years, I received a bug report from a vakthesis user. If he uses the `bibset` environment with hyperref loaded, then any `\cite` command is not correctly hyperlinked.

The reason is that recent versions of hyperref do not define `\hyper@warn` anymore. The command `\Hy@WarningNoLine` is defined instead. Hence the ⟨*true branch*⟩ in that code is not executed at all.

Obviously, the quick patch is to restore the definition in the document preamble:

```
\@ifundefined{hyper@warn}
  {\let\hyper@warn\Hy@WarningNoLine}
  \relax
```

In the new version of vakthesis, the hyperref check is also fixed.

However, checking if a package is loaded inside of a command/environment is a bad thing anyway. To make more robust software, I should prepare two versions of a command/environment (with and without hyperref) and then choose the corresponding version.

2.5 Overwriting and overloading

I started to develop the vakthesis bundle in approximately 2003. I was a PhD student at that time. My experience with TEX and LATEX was very limited. Sure, I used LATEX for preparing my papers, slides, etc.; but I did not try LATEX programming. I should say also that Internet access was unstable and expensive at that time.

As a result, I did not find any suitable templates or LATEX classes for thesis typesetting compatible with Ukrainian requirements. I thought the situation required development of a LATEX class that I needed myself.

At some point, I decided to start from the standard report class and modify it according to my needs. It seemed that overwriting of an existing class is a better approach. There exists a ready-to-use class that almost complies with my requirements. I just need to patch some parts of the code that do not agree with the requirements.

However, in fact, it is not easy to maintain such a new class. First of all, I should look at report class development and update my code. At that time, I thought that changes of the standard LATEX classes are rare.

Moreover, careless overwriting may cause problems. For example, in the report class, the command `\part` typesets part headings on separate pages. But the vakaref class uses this command to typeset structural parts of summary; and they are typeset as usual headings. So I removed any `\newpage` commands and stopped.

One day a user encountered a problem when a structural part heading occurs on the last line of the page. As a result, a page break appears between the heading and the following text. This is unwanted behavior, of course, and fixing of the command is required.

This is one of the reasons why modern classes mon2017dev and mon2017dev-aref use another approach. They are designed as a "level" above the traditional classes vakthesis and vakaref respectively. They load these base classes and then redefine some commands and environments.

Unfortunately, overloading may cause problems too. Because the casus package (see Section 2.3) works with Cyrillic letters directly it should be loaded after the inputenc package. So, in the vakaref class, there is a corresponding line:

```
\AtBeginDocument{\usepackage{casus}}
```

Since mon2017dev-aref is a level above the vakaref class, the following line

```
\LoadClass{vakaref}
```

is in the mon2017dev-aref.

For the mon2017dev-aref class, some modifications in casus are needed. So this class loads a modified version of this package:

```
\AtBeginDocument{\usepackage{casus2017dev}}
```

Of course casus2017dev cannot work without casus. But we have the following chain. Firstly, the mon2017dev-aref class loads the vakaref class, then vakaref adds casus to the `\AtBeginDocument` hook. Later the mon2017dev-aref class adds casus2017dev to the `\AtBeginDocument` hook. As a result, in a proper place, casus is loaded, and then casus2017dev is loaded.

This colossus with feet of clay did its excellent work... till one day when it fell. A user is in a slight panic: declension does not work in his summary, and the cover page of the summary is full of nonsense! Oh, and he should submit his thesis and summary today! More interesting, I do not see this problem on my system.

The reason is in a new release of LaTeX. In version 2020-10-01, a general hook management system was provided [3]. This affects standard hooks defined by the command `\AtBeginDocument` too.

I am not sure if I understand correctly what exactly happens. I suppose that, since casus and casus2017dev are loaded in the different classes, we have that they are added to hooks with different labels. As a result, either code is executed in changed order or the second `\AtBeginDocument` just executes code instead of adding to the hook. The visible result is that casus2017dev is loaded before casus. It cannot work in this situation.

It is easy to fix this problem. It is enough to add the line

```
\RequirePackage{casus}
```

to the casus2017dev package.

Clearly, such many-level overloading may be a real headache for users as well as for maintainers.

Taking into account the above-mentioned problems, I do not know now what is the best way to develop a new class: copy an existing class and rewrite some parts of it, or load a base class and redefine the commands/environments.

3 Current status

3.1 Two separate modules

Now the vakthesis bundle consists of two separate, almost independent, modules: traditional vakthesis classes and modern mon2017dev classes.

I started development of mon2017dev as a separate module to keep the vakthesis module stable and harmless for users. In some aspects, modern requirements by MON differ essentially from traditional requirements by VAK. Sometimes they are unclear and ambiguous. So mon2017dev classes are intended for clarifying development objectives and obtaining feedback from users.

However this separation causes some problems today. In particular, separate installation of two modules is more complicated for users. Documentation is also divided between two modules.

For me as the maintainer, development and maintenance of two modules also requires additional effort. For example, problems similar to the case with casus/casus2017dev (see Section 2.5) are mostly caused by this separation.

3.2 Alternatives

Some alternative solutions for typesetting a thesis in Ukraine exist.

First of all, I would like to mention a long-standing dissert class by Andrew Martovlos. This class is based on the report class and size14 class option and published in 2002.

Unfortunately, the website where dissert was initially posted does not exist anymore. Also, there exist a number of derivatives of dissert by now. Probably they are even mutually incompatible. One of them with the name dissert_new is published at the Linux.org.ua forum [4].

I have a copy of an Opus class by Andrii Semenov. Sergei Sharapov sent me this copy in 2009 and said that this class is used in Bogolyubov Institute for Theoretical Physics of the National Academy of Sciences of Ukraine. However I do not know its current status.

Another class is recently published at the above-mentioned forum too [2]. This is ukrainethesis class by Kostiantyn Hermash. He used it to prepare his PhD thesis.

So, users that are unsatisfied with vakthesis LaTeX classes may choose another solution. Nevertheless, one considerable difference between vakthesis and other classes is that I actively maintain vakthesis and answer user questions about it.

3.3 Users of vakthesis

Despite the fact that some unsolved problems exist and these problems can be critical for some users, there exists some interest in vakthesis.

Many people use these classes to typeset their doctoral theses belonging to various fields: biology, computer science, mathematics, physics, etc. Also some students adapt the classes for their student qualifying works although vakthesis is not intended for this task.

A friendly community of vakthesis users is gathered at the Linux.org.ua forum: linux.org.ua/. Any user can ask questions here and receive support from the community.

Recently the Doctor Barbarus Services company offers commercial support for all users that need it: sites.google.com/view/drbarbarus/.

4 Future plans

I hope the following changes will resolve some complicated problems mentioned in the above sections as well as make the vakthesis bundle more convenient for users and developers.

4.1 Combine vakthesis and mon2017dev

The modern classes mon2017dev are quite stable software already. So they can be combined with the base vakthesis classes. However, this is not just mechanical work if we want to keep compatibility.

4.2 Make vakthesis fully UTF-8 compatible

Full rewriting of the casus package is the main problem in this direction. This is an important issue because the UTF-8 encoding is a standard in the modern world.

4.3 Provide more documentation

In particular, the installation process should be described with more details for non-experienced users.

More useful and user-friendly example files are also needed. Some information related to real persons should be removed from the example files. For examples with traditional vakthesis classes, I used some parts of my thesis. The modern classes mon2017dev are illustrated by examples from the MON style guide that contain real information too.

4.4 Upload to CTAN

I did not upload the vakthesis bundle to CTAN earlier because I believed that it is not yet sufficiently stable software.

However I hope that above-mentioned changes will lead vakthesis to be more mature and usable.

Moreover having the software available at CTAN under a free license will also make it available in the main TEX distributions. So it will become easily installable.

4.5 Use version control system

I did not use any version control system earlier. But now I understand its importance for development and maintenance.

Later I will upload code to GitLab, GitHub, or my own server (to be decided).

This may help other developers continue this project if I will not be able to maintain it for some reasons. I am in Ukraine now; so such reasons can occur any day.

5 Acknowledgments

I am grateful to Barbara Beeton and Karl Berry for their careful reading the manuscript and improving the presentation and English wording.

This paper was prepared in Ukraine during the russian invasion. I sincerely appreciate the Armed Forces of Ukraine's heroic efforts to defend my country.

References

[1] H. Adriaens, U. Kern. xkeyval — new developments and mechanisms in key processing. *TUGboat* 25(2):194–199, 2004. tug.org/TUGboat/tb25-2/tb81adriaens.pdf

[2] kostiantyn.hermash. Message at the Linux.org.ua forum. linux.org.ua/index.php?topic=689.msg204167#msg204167

[3] LaTeX Project Team. LaTeX2ε news 32, Oct. 2020. www.latex-project.org/news/latex2e-news/ltnews32.pdf

[4] sampleplasma. Message at the Linux.org.ua forum. linux.org.ua/index.php?topic=689.msg92080#msg92080

[5] TeX FAQ contributors. Multiple bibliographies? texfaq.org/FAQ-multbib

[6] TeX FAQ contributors. Separate bibliographies per chapter? texfaq.org/FAQ-chapbib

[7] Current homepage of the vakthesis project. www.imath.kiev.ua/~baranovskyi/tex/vakthesis/

[8] J. Wright, C. Feuersänger. Implementing key–value input: An introduction. *TUGboat* 30(1):110–122, 2009. tug.org/TUGboat/tb30-1/tb94wright-keyval.pdf

◇ Oleksandr Baranovskyi
Doctor Barbarus Services & Institute of Mathematics of the National Academy of Sciences of Ukraine
ombaranovskyi (at) gmail dot com
https://sites.google.com/view/drbarbarus/