LaTeXercise - *An exercise sheet generator*

Julian Maicher, Alexander Morast
Universität Paderborn, Supervisor: Prof. Dr. Uwe Kastens
Generating Software from Specifications
Email: jmaicher@mail.upb.de, alm@mail.upb.de

1 Introduction

Our goal is to create a simple and intuitive abstraction layer on top of LaTeX to create high-quality exercise sheets. We want to enable our users to take advantages of LaTeX without them having to write LaTeX. Therefore we design a domain-specific language (DSL) to describe exercise sheets and we implement a compiler, *LaTeXercise*, which will generate the proper LaTeX source from our DSL.

In section 2 we will outline the anatomy of exercise sheets which will be used in section 3 as foundation to design an exercise sheet DSL. In section 4 we show how the output of the *LaTeXercise* compiler will look like. Finally we will give future prospects on how we might improve the DSL and the compiler and how me might use it once the basic features are implemented.

2 Anatomy of exercise sheets

This section describes the domain of exercise sheets and will be used to identify the requirements for our DSL. We do not aim to cover all edge cases but we do want to provide a good foundation which will be sufficient for the most general case.

An exercise sheet will be published in the context of a *lecture* which is held by a *lecturer* in a specific *term* at an *educational institution*. It has a *release date* and may has a *submission date* and an *issue* number. An exercise sheet consists of at least one *task* which may include multiple *subtasks* . Both, tasks and subtasks have a textual description. Tasks are numbered and subtasks are either numbered or itemized. Typically task numbering starts from 1) but could also start with an arbitrary *startnumber*.

3 Design of the exercise sheet DSL

As we see in section 2, an exercise sheet consists of predefined metadata and different tasks. Thus, the DSL requirements can be divided into two main parts: 1) the description of the metadata and 2) the description of exercise tasks.

3.1 Metadata

In order to describe the metadata of an exercise sheet, a simple key-value mapping is sufficient. Therefor we've choosen the following syntax for our DSL: `<key>:<value>`. A list of the predefined metadata keys is shown in Table 1.

Name	Type	
lecture	`String`	
lecturer	`String`	
institution	`String`	
term	`String`	
release	`Date`	
submission	`Date`	*optional*
issue	`Integer`	*optional*
startnumber	`Integer`	*optional*

Table 1. Metadata of an exercise sheet

The metadata definitions have to be given in a metadata definition block which must be started and terminated by a line starting with one or more dashes (-). Metadata definitions inside the block must be separated by new lines. The metadata definition block should be given at the beginning of the file. Section 3.3 gives an example of a metadata definition block.

3.2 Exercise tasks

When describing an exercise sheet, our users will spend most of their time describing the specific exercise tasks and hence this part of the DSL should be as easy and convenient as possible.

Tasks consist of a textual description and either numbered or itemized subtasks. Subtasks again have a textual description. As both, tasks and subtasks have a textual description, they can be represented by text blocks. Our DSL still needs ways to distinguish between multiple tasks and to express task hierarchy. Therefor we've decided on the most basal and intuitive ways we could think of: new lines and indentation.

In order to describe the first task, our users can simply write down the task description as plain text. When they want to describe a new task, they just need to start a new line. If the new task is a subtask, the new line has to be indented. Indented lines either start with a *whitespace* or *tab* character. This is a feature because users may have set *tabs* to be outputted as multiple *whitespace* characters in their text editors and we don't want to be optionated about that.

Subtasks are numbered by default. In order to express that they are itemized, the indented line can begin with a dash (-). The following section gives an example of task descriptions.

3.3 Example

Listing 1 gives a concrete example of our exercise sheet DSL. The metadata definition block is given at the beginning of the file. The first task has two numbered subtasks. The second task has three itemized subtasks and the third task does not have any subtasks at all. Because no *startnumber* is given in the metadata, the task numbering will start from 1) by default.

```
---
lecture: Generating Software from Specifications
lecturer: Prof. Dr. Uwe Kastens
institution: University of Paderborn
term: Summer term 2012
issue: 3
release: 2012-05-15
submission: 2012-05-21
---

This is the description of the first task which consists of
   numbered subtasks

  This is the first subtask. In order to express task hierarchy
     the new line has to start with either a tab or whitespace

  This is the second numbered subtask


This is the second task which has itemized subtasks.

  - When subtasks start with a dash they will be itemized and not
     numbered
  - This is another itemized subtask
  - The third itemized subtask is the last


This is the last task which does not have subtasks
```

Listing 1. Concrete example of our exercise sheet DSL

4 LaTeX output

The *LaTeXercise* compiler takes as input the exercise sheet DSL defined in section 3 and compiles it to a proper LaTeX source file. In order to simplify the output, we define a document class, *LaTeXercise*, which is highly inspired by a document class created by Rolf Wanka[1] at the University of Paderborn. This document class will provide custom LaTeX commands to set metadata and to describe tasks.

4.1 Example

Listing 2 gives an example output of the *LaTeXercise* compiler for the input given in Listing 1.

[1] wanka@uni-paderborn.de

```
\documentclass{LaTeXercise}
\begin{document}

% ****************************
% metadata definitions *******
% ****************************

\lecture{Generating Software from Specification}
\lecturer{Prof. Dr. Uwe Kastens}
\institution{University of Paderborn}
\term{Summer term 2012}
\issue{3}
\release{2012-05-15}
\submission{2012-05-21}


% ****************************
% task descriptions **********
% ****************************

\begin{task}
This is the description of the first task which consists of
   numbered subtasks

\begin{enumerate}[a)]
  \item This is the first subtask. In order to express task
     hierarchy the new line has to start with either a tab or
     whitespace
  \item This is the second numbered subtask
\end{enumerate}
\end{task}

\begin{task}
This is the second task which has itemized subtasks

\begin{itemize}
  \item When subtasks start with a dash they will be itemized and
     not numbered
  \item This is another itemized subtask
  \item The third itemized subtask is the last
\end{itemize}
\end{task}

\begin{task}
This is the last task which does not have subtasks
\end{task}

\end{document}
```

Listing 2. Example output of the *LaTeXercise* compiler

5 Future prospects

Once the basic features described in the previous sections are implemented, we might improve the DSL and the compiler with more functionality.

5.1 Text formatting

We plan to extend the exercise sheet DSL with basic text formatting options like *italic*, **bold** and `code`. Furthermore we think about adding support for links and footnotes[2] to the DSL. Tabel 2 shows a proposal of how the fomatting support in our DSL could look like. The proposed DSL-syntax conforms to the widely-used *Markdown*[3] syntax.

Formatting option	Input (DSL)	Output (LaTeX)
italic	*italic*	`\textit{italic}`
bold	**bold**	`\textbf{bold}`
`code`	`` `code` ``	`\texttt{code}`
links	[label](http://..)	`\href{http://..}{label}`
footnote[4]	_(text)	`\footnote{text}`

Table 2. Future prospects for the DSL to support formatting options

5.2 Images

Adding support for images could be very useful to a lot of users. We propose the following syntax for the exercise sheet DSL: `![caption](path/to/img.jpg)`. This syntax conforms to the *Markdown* syntax for images. A drawback from this syntax is, that images can't be referenced from within text. Thus, we may change the syntax when adding image support.

5.3 Mathmatical formulars

A common task when creating exercise sheets is to embed mathmatical formulars. The LaTeX math environment already provides a good DSL for mathmatical formulars. We may support the embedding of these formulars by adding the following syntax: `$LaTeX math formular$`.

5.4 *LaTeXercise* cli

The idea of the *LaTeXercise* cli is to make the use of LaTeX completely invisible to the user. The exercise sheet DSL is an abstraction of the LaTeX syntax but the user still has to deal with the LaTeX source created by the *LaTeXercise* compiler.

The *LaTeXercise* cli provides a command-line interface and will immediately compile our exercise sheet DSL into a PDF. The example in Listing 3 illustrates the usage of the cli.

[2] This is a footnote

[3] `http://daringfireball.net/projects/markdown/syntax`

```
$ latexercise sheet01.exs
Compiling to latex...done
Compiling to pdf...done

sheet01.pdf has been created successfully.
```

Listing 3. Example usage of the *LaTeXercise* cli