Convergent Publishing Workflow for Online Classes

Teaching Materials from Meta Formats

https://doi.org/10.3991/ijet.v16i11.21433

Alaric Hamacher
Kwangwoon University, Seoul, Republic of Korea
stereo3d@kw.ac.kr

Abstract—Online teaching in 2020 forced many educators to adopt new teaching methods. Instead of working in the classroom with handout and oral presentations, online teaching requires new teaching materials and documents. These are usually created in different formats with different software and is often redundant. The present paper proposes a research on workflows and practical applications to streamline the publishing process by proposing authoring in a meta data format and publication of convergent teaching material from a single document. The purpose of this research is to improve the quality of education by reduction of redundant workflows in the creation process of teaching materials.

Keywords—Publishing, Online Classes, Teaching Materials, XSLT

1 Introduction

2020 brought big changes in education. The fear of contamination due to the pandemic forced many educational institutions to change curricula fundamentally and move teaching as well as communication and assessment online [1]. As a consequence, not only teaching practices changed, but the whole communication with students, printouts and publications as well. Multiple recommendations have been published with examples from mostly commercial software services [2]. While many services have discovered educators as a new target group, the present research proposes free and open-source tools and workflows that can be achieved without subscription fees. Some teachers use textbooks, others often create their own new teaching materials [3]. These materials can have different kinds and shapes. Some examples are: handouts, keynote slides, presentation backgrounds, textbooks. They often have very similar content in a slightly different layout. The following section gives a brief overview of some of the most popular publication targets.

1.1 Keynote presentations

Keynote presentations are very often used in teaching. In many cases they have replaced traditional teaching based on the use of chalkboards [4]. The presentation usually consists of multiple slides showing keywords and topics of the current lesson. In addition, these presentations may contain images, videos and sound.

Educators often use dedicated software for the creation of keynote presentations. This kind of software is usually part of a so-called office package. Popular examples are Microsoft PowerPoint, or LibreOffice Impress [5]. Also popular among Apple users is the software Keynote. The software Latex can also be powerful tool for creating PDF presentations especially when programming and linking inside documents is required.

1.2 Student handouts

Student handout designs can be very similar to keynote presentations. They can have reduced sizes. Some educators print the slides of their presentation. Others prefer to downsize the slides and print multiple slides on one page. They usually also contain space where students can add notes. Figure 1 shows different types of handouts. Handouts are usually a subtype of the keynote presentations. The above-mentioned software for keynote presentation has in all cases the special future for printing out handouts.

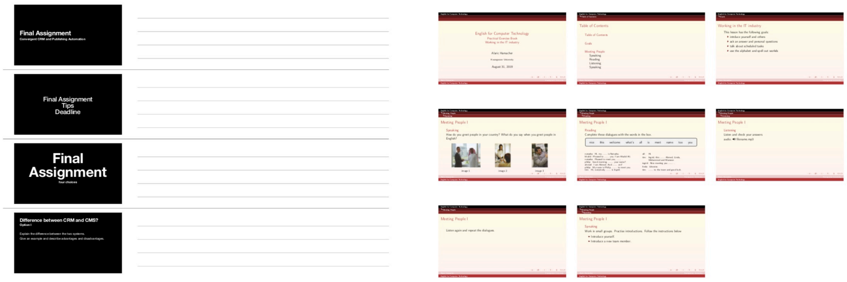

Final Assignment
Convergent CRM and Publishing Automation
Final Assignment
Tips
Deadline
Final Assignment
Difference between CRM and CMS?
English for Computer Technology
Table of Contents
Meeting in the IT industry
Meeting People I

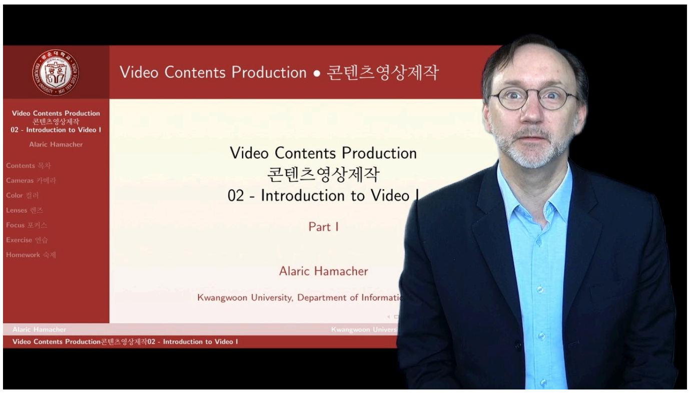
Fig. 1. Student handout with notes (left), multipage handout (right)

1.3 Textbook

The textbook is an important publication target. Some textbooks can serve as a replacement for a class. They contain in written form the introduction and text teacher would usually explain orally in the classroom. Textbooks are also created to study alone. Therefore, text and explanations are not only limited to keywords but contain more elaborate descriptions. Descriptions in the textbook are very similar to the speech presented in front of the class. They are also very similar to the lecture dialogues in an online web cast. Large parts of textbooks can serve as the transcript for a class [6].

1.4 Web casts

Online seminars and web casts usually contain some graphical elements in front and behind the presenter and special multimedia software for authoring can be necessary [7]. In certain cases, teachers can use the same keynote presentation as they would use in the classroom. However, there are also many cases, where it is inappropriate. When the teacher is in the same picture as the presentation, overlappings can occur and important keywords can be unreadable for the viewer. Especially when multiple cameras are used and the size of the presenter changes constantly. Figure 2 shows a keynote publication as web cast background.



Video Contents Production • 콘텐츠영상제작

Video Contents Production
콘텐츠영상제작
02 - Introduction to Video I

Alaric Hamacher

Contents 목차
Cameras 카메라
Color 컬러
Lenses 렌즈
Focus 포커스
Exercise 연습
Homework 숙제

Video Contents Production
콘텐츠영상제작
02 - Introduction to Video I

Part I

Alaric Hamacher

Kwangwoon University, Department of Informati

Alaric Hamacher
Kwangwoon Univers
Video Contents Production콘텐츠영상제작02 - Introduction to Video I

Fig. 2. Keynote as web cast background

1.5 Exams

Exams and quizzes can often have similarities with examples or exercises presented in class. In this case producing an example for a textbook or creating an exercise in the workbook is very similar to creating part of a quiz or an exam. In case of learning management systems (LMS) often multiple-choice questions are used. These offer a great potential for reusability with only very slight modifications. Software for online exams is usually integrated in learning management systems. The students' results and performance are stored in learning record stores (LRS). This allows the educators to follow the students' progress and results in real-time. Many other systems exist to create exams and quizzes off-line [8].

The above-mentioned publishing targets are examples. There are more documents that can be used for teaching and exposing all would go beyond the scope of an article. Since using existing textbooks is not always possible, teachers often rely on their own materials and are facing increasing challenges in preparation.

2 Problem

When educators create their own teaching materials, they have to target different publishing outputs and often use different software for each. An attempt to classify some of the existing tools during the Covid-19 pandemic has been achieved in the survey by Tawafak et al. [2]. Most modern publishing tools provide solutions for a particular publishing target. However, in many cases the layout is so closely connected to the content that it can become difficult to reuse teaching materials made for one publishing target for another. For example: explanatory texts and descriptions for a textbook are often composed using text processor. The text processor is generally made for documents in vertical format. All text, image, paragraph formats and layouts are adapted to this vertical page format. It is not easy to change the layout and content to match the design of the horizontal format of a keynote presentation. In many cases content has to be rewritten or copied from one software to another. Common problems of this process are:

- Difficulty keeping track of changes
- Impossibility to reuse content
- Difficulty to adapt content for new publishing targets
- Heavy workload for creating new classes

When authoring teaching materials, it becomes a complex issue to keep track of changes: a modification in one teaching document will not automatically be reflected in another. A lot of time is spent to recreate similar documents in different layouts. Almost every document represents a dead end because in most cases this document has only one single application and cannot be used in another target. It is very difficult to reuse content elements. Adding new publishing targets such as a HTML5 e-learning platform or video online classes in the workflow represent additional challenges. For many new classes a large amount of time is dedicated to authoring and design of new course materials. Education quality could improve if more time could be spent in work on the content of the class instead of redundant time-consuming creation of teaching materials.

The following section shows a brief overview and evaluation of present technologies for document conversion and multiple target publishing. The fourth section of this article will propose a workflow method based on the selected tools. The final section of this paper draws the conclusion about the presented workflow and indicates fields for further research.

3 Present Technologies

Document formatting and conversion are important topics in publishing. Multiple solutions exist already for specific applications. This section focuses on a selection of application examples and describes their specificity and evaluates possible advantages and disadvantages for the creation of online course materials.

3.1 Office automation (Databases)

Publication of data from separate layout can often be achieved with office automation. Prominent examples are serial letters and mail merge [9]. The procedure consists usually of creating two different files: one is a database, the other one is a publishing target template. The database can regroup a multitude of fields in records. The text document is generally a letter or catalogue. Every time a new page is created placeholder values are updated with matching values of the database. Different publishing targets can be achieved by using a variety of letter templates [10]. The advantage of this procedure is that it is very easy to use, most office software has this kind of publishing feature already built in. However, the disadvantage is that the database or table is mostly made of very similar data. Complex documents with more detailed structures are difficult to achieve with this method.

3.2 Shareable Content Object Reference Model (SCORM)

The SCORM model is a set of specifications and rules specifically targeted for e-learning. It is a result of the advanced distributed learning initiative initiated by the office of the United States secretary of defense [11]. Created in 2004, SCORM defines sequences and rules for learning content and interaction between user and the software[12]. The specifications describe how an application should behave. The teaching material is stored in XML files. This universal file format makes it easy to share the educational content among different LMS. SCORM is a standard in e-learning. It assures that learners can have the same learning experience within different learning applications. The advantage of SCORM is the standard for interoperability and reusability of educational content. However, the standard does not define exactly output to the user. This has to be implemented by each software individually. The complexity of the standard can be an obstacle for quick content creation. The following XML code shows the manifest definition in SCORM.

```
<manifest xml:base="Course/">
  <organizations>
    <organization>
      </item identifier="ID1" identifierref="R_ID1">
    </organization>
</organizations>
...
</manifest>
```

3.3 Pandoc (Markup conversion)

The conversion between different file formats is the purpose of pandoc. It is a software tool specially designed to convert markup languages [13]. Markup formats are very popular. They can be used to easily create documentations, slideshows, HTML, XML, office documents or e-books. In principle certain tags or characters structure the content. This allows software to create the appropriate formatting. Pandoc translates a markup language created by one software to another markup language used by another software. The advantage of this tool is that many conversion filters already exist. However, as markup languages can become very complex sometimes. Consequently, not all features are supported. It also does not create multiple different documents from one single source.

3.4 Latex (Document classes)

Latex is a very versatile tool for typesetting. It is one of the oldest programs for publishing. Markups are used to structure documents [14]. Document classes describe features and layout documents. Latex makes it possible to create different publishing targets from the same content by switching document classes [15]. The *article* class aims at scientific publications, the *beamer* class aims at the creation of keynote presentations. Many additional packages allow users to extend features of the software. The advantage of Latex is that is possible to create almost any type of printed document. It can also be used to create multiple publication targets from one document. This makes it very useful for the creation of various teaching materials. However, handling the software has a steep learning curve and document creation can seem not as intuitive for authors accustomed to live previews and graphical user interfaces.

3.5 HTML + CSS (Style sheets)

HTML and CSS are the basis of most content on the Internet [16]. While HTML uses a markup language to structure a document, design and layout of the document structure are usually stored apart in a stylesheet. Stylesheets allow to design the documents to best match the proportions of the devices on which they are displayed: e.g., a computer screen, a tablet or a mobile device [17]. Despite this flexibility, they are not specifically designed to create multiple publishing targets from the same document.

3.6 XML stylesheet transformations (XSLT processor)

In context of content creation, the XML stylesheet transformation processor can play an important role. The processor is an engine that takes two inputs: one is a data file in XML format [18]. The other one is a style sheet file also written in XML format. The stylesheet contains a set of instructions in the form of templates. These are applied to the tags in the input XML document. XML stylesheet transformation language (XSLT) allows very complex structures, it can include conditional logic and branching [19]. The XML stylesheet can be used to create a new XML document on the basis of XML data. It can also be used to produce any kind of text file. This makes it a very interesting candidate for creating multiple types of documents in very different markup language formats. The advantage of XML stylesheet transformation is its very wide field of application. This is why this tool can be a very good option to achieve the role of multiple target publication. Table 1 summarizes the previously discussed technologies.

Table 1. Overview of publishing methods

Method	**Specialty**	**Publishing targets**				
		Keynote	***Handouts***	***Quiz***	***Webcast***	***Textbook***
Office	Document creation	x	x			x
SCORM	E-learning exchange			x		
PANDOC	Markup conversion	x				x
LATEX	Print document creation	x	x	x		x
HTML+CSS	Web document creation			x	x	
XML+XSLT	Text generator	x	x	x	x	x

4 Proposed Method

This section focuses on two main topics of a workflow solution: the first one is the proposition of a metadata structure in XML. This format is robust and humanly readable as well as readable by computers. XML is an established format and many tools exist to create and validate documents. Examples will demonstrate how a simple structure can be created. As the demand for additional options grows, more features can be added to the format in in the future.

The second topic in this section will demonstrate how stylesheets can be applied using XSLT and template matching. A practical example will show how two different stylesheets are used to create a textbook as well as a keynote presentation that can be used in background for online teaching. These documents will be derived from the same metadata file.

4.1 Creation of the meta format

Creation of the meta format follows the identification of recurring structures in a course: a course can be compared to a book. It can run over a full semester. The semester usually has 15 weeks. And every week can consist of one or more units. This can correspond to the chapters of a book. A chapter can contain different sections. These sections also contain recurring elements in the lesson: introduction, example, exercise, theory, etc. The names of these sections depend on the class subject. Each section can contain different tasks adapted to the subject of the class. The meta format should be easy to create and to maintain. The structure should be simple and support future development. As example the following program code represents the structure of an English language book. The proposed meta format allows creating a textbook, keynote presentation and student handouts at the same time.

```
<chapter id="1">
  <chaptertitle>English for IT</chaptertitle>
  <chaptergoals>
    <goal>introducing yourself and others</goal>
    ...
  </chaptergoals>
```

The code shows the description of the first lesson of the English course. The lesson is enclosed with a *chapter* tag. Directly at the top of the chapter additional tags give more information about the lesson: the *chaptertitle* describes the title of the lesson. The tag *chaptergoals* is a placeholder for different *goal* elements that can describe the lesson's objective.

Tasks with modules: Below the chapter information, the lesson contains different sections. There can be as many sections as units are needed in the chapter. Each section starts with a section name. In this case the first lesson of the English class is the introduction starting with a speaking exercise called "meeting coworkers". The section consists of a number of tasks. The attribute *module* allows to specify the kind of task that is part of the section. Each module refers to a certain style that is used along the textbook or in keynote presentations. In this example a small image gallery should give inspiration for the speaking exercise. The illustration tag contains a list of assets to be inserted when needed in the targeted output. The subsequent code shows how the gallery module can be implemented:

```
  <sectionname>Meeting Coworkers</sectionname>
       <task module="gallery">
    <taskname>Speaking</taskname>
    <taskdescription>How do you greet your
friends?</taskdescription>
      <illustration><image><file>ch01f01a.png</file>...
```

Match the Blanks: *Match the blanks* is an exercise where students have to fill up empty spaces in dialogue. This type of exercise is used in many different lessons. It can be part of the workbook; it can be displayed in the keynote in front of the class and it can also become part of an examination. In this case also multiple publishing targets can be prepared: one part will be the exercise with empty spaces, another part will display the solution with the correct answers.

```
<task module="matchblanks">
  <taskname>Reading</taskname>
  <taskdescription>Complete these dialogues with the
                 words in the box.</taskdescription>
    <dialog>
      <dialogline>
        <actor>Natasha</actor>
        <text>Hi, my <blank>name</blank>
              is Natasha.</text>
      </dialogline>
```

The program code shows how this module can be implemented to display a dialogue with different lines and blanks.

4.2 XSLT Stylesheet

The transformation of the metadata input file into a publishing output, can be achieved by using a stylesheet. The stylesheet consists of templates matching the various XML tags and attributes of in the XML input file [19]. One stylesheet is used for each publishing target. In order to keep better track of changes, to increase the readability and maintaining the stylesheet is constructed with a principal main document and importing constant and modules from separate library files. These separate library files contain particular layout of the modules: one module can describe the output in a keynote presentation, while another module describes how the same content is to be presented inside a text book. Recurring variables can be achieved using constants for global data such as course title and author as shown in the following code example:

```
  <xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"   ver-
sion="1.0">
    <xsl:variable name="title" select="'English for Com-
puter Technology'"/>
    <xsl:variable name="subtitle" select="'Practical Ex-
ercise Book'"/>
    <xsl:variable name="author" select="'Alaric
Hamacher'"/>
    ...
  </xsl:stylesheet>
```

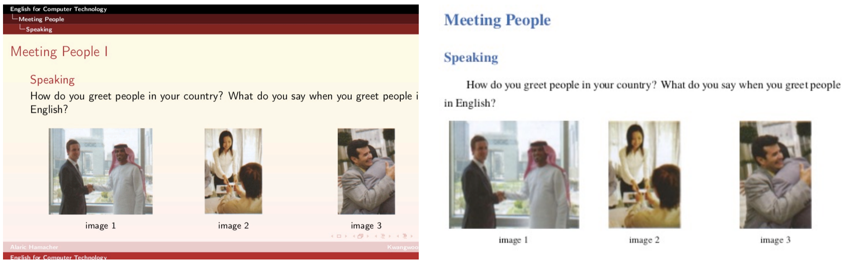
Module-Gallery: The first template matching example demonstrates the *gallery* module with image display. For this the template matches the *task* element in the XML input file containing the attribute *gallery*. The following code creates Latex code for a new document:

```
  <xsl:template match="task[@module='gallery']">
      <xsl:variable name="imagecount">
      <xsl:value-of select="count(illustration/image)"/>
    </xsl:variable>
    <xsl:variable name="imagefraction">
      <xsl:value-of select="round(1 div $imagecount *
100) div 100"/>
    </xsl:variable>
    <xsl:text>\begin{figure}[ht]&#10;</xsl:text>
    <xsl:text>\captionsetup[subfigure]{labelformat=empty}
&#10;</xsl:text>
    <xsl:text>\centering&#10;</xsl:text><xsl:text>\hfill
&#10;</xsl:text>
    <xsl:for-each select="illustration/image">

  <xsl:text>\begin{subfigure}[b]{</xsl:text><xsl:value-of
        select="$imagefraction"/>
      <xsl:text>\linewidth}&#10;</xsl:text>

  <xsl:text>\centering\includegraphics[height=4cm]{</xsl:
text>
      <xsl:value-of se-
lect="file"/><xsl:text>}&#10;</xsl:text>
      <xsl:text>\caption{</xsl:text><xsl:value-of se-
lect="caption"/>
      <xsl:text>}\end{subfigure}%&#10;</xsl:text>
    </xsl:for-each>
    <xsl:text>\end{figure}&#10;</xsl:text>
  </xsl:template>
```

The purpose of this module is to align a random number of images evenly in the presentation target: First the available space for each illustration is calculated from the page width. Then, the template creates a Latex markup with a caption environment to display the images. Figure 3 shows the output of this code in two different document classes:


English for Computer Technology
└Meeting People
└Speaking

Meeting People I

Speaking

How do you greet people in your country? What do you say when you greet people i English?

Meeting People

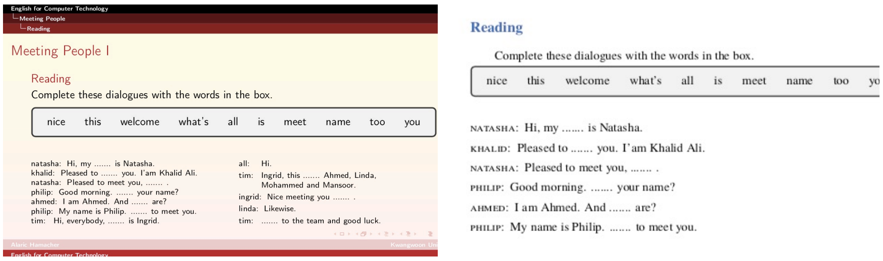
Speaking

How do you greet people in your country? What do you say when you greet people in English?

Fig. 3. Proposed gallery module in Latex beamer class (left) and book class (right)

4.3 Module: Match the blanks

Similar to the *gallery* module, the *match in the blanks* module creates the layout for an exercise in the class or for the textbook. The correct answers of this exercise need to be omitted. Instead, the valid answers are regrouped in the grey box on top of the dialogue. In order to achieve this, the matching template performs two iterations: a first one to extract the correct answers from the dialogue. And a second one to create a dialogue with blank spaces. An additional template can be used at the end of the book to display the correct solutions. Figure 4 shows how the exercise is rendered as a two-column dialogue in the keynote presentation, and in single-column layout in the book:


English for Computer Technology
└Meeting People
└Reading

Meeting People I

Reading

Complete these dialogues with the words in the box.

nice this welcome what's all is meet name too you

natasha: Hi, my .... is Natasha.
khalid: Pleased to .... you. I'am Khalid Ali.
natasha: Pleased to meet you, ...
philip: Good morning. .... your name?
ahmed: I am Ahmed. And .... are?
philip: My name is Philip. .... to meet you.
tim: Hi, everybody, ... is Ingrid.
all: Hi.
tim: Ingrid, this .... Ahmed, Linda, Mohammed and Mansoor.
ingrid: Nice meeting you ....
linda: Likewise.
tim: .... to the team and good luck.

Reading

Complete these dialogues with the words in the box.

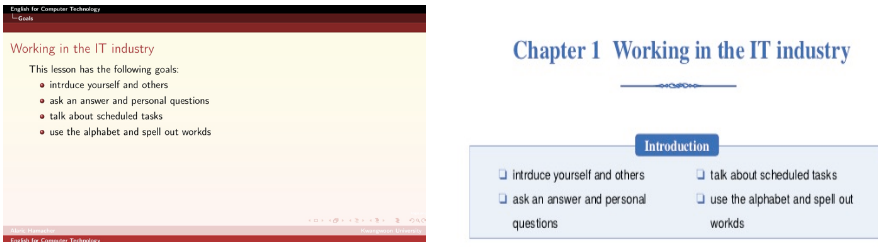
nice this welcome what's all is meet name too yo

NATASHA: Hi, my .... is Natasha.
KHALID: Pleased to .... you. I'am Khalid Ali.
NATASHA: Pleased to meet you, ...
PHILIP: Good morning. .... your name?
AHMED: I am Ahmed. And .... are?
PHILIP: My name is Philip. .... to meet you.

Fig. 4. Match the blanks module in Latex beamer class (left) and book class (right)

4.4 Module: Instructions

The example in Figure 5 shows different display methods of the lesson goals in a keynote and in a textbook. While the keynote presentation uses bullet points and slides, the textbook can use text decorations, highlighted boxes and different colors.



English for Computer Technology
└Goals

Working in the IT industry

This lesson has the following goals:

- intrduce yourself and others
- ask an answer and personal questions
- talk about scheduled tasks
- use the alphabet and spell out workds

Alaric Hamacher — Kwangwoon University
English for Computer Technology

Chapter 1 Working in the IT industry

Introduction

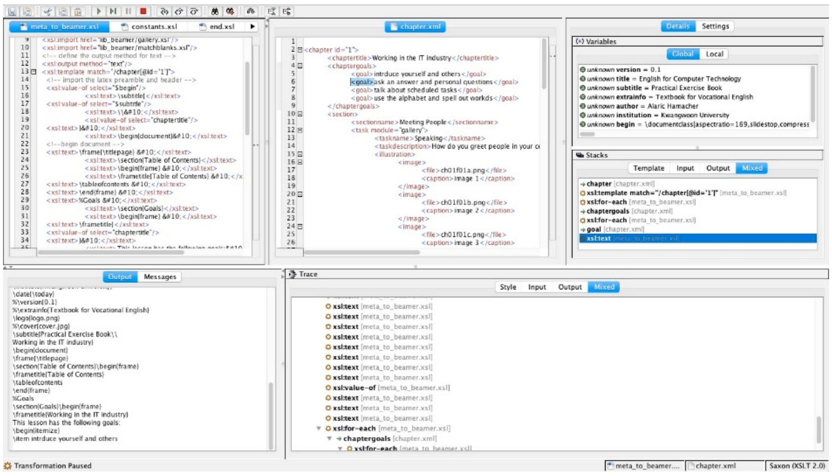
❑ intrduce yourself and others
❑ ask an answer and personal questions
❑ talk about scheduled tasks
❑ use the alphabet and spell out workds

Fig. 5. Display of chapter goals in Latex beamer class (left) and book class (right)

All individual display modules are created in separated files and imported in a master stylesheet. One master stylesheet is created for each intended target output (keynote, textbook, web cast background).

4.5 Running the transformation

In order to perform the transformation from the input XML file containing the teaching content with a stylesheet an XSLT processor is required. Commercial and open source solutions are available. Figure 6 shows how the complete transformation process can be monitored using the XSLT debugger software xngr [20]. The window on the top side contains the stylesheet transformation file and the XML input file. On the lower left side, the generated latex output content is visible. The global variables are visible on the right side. The transformation can be executed in single-run or line-by-line during development to verify and adapt the output.



```
<xsl:import href="lib_beamer/gallery.xsl"/>
<xsl:import href="lib_beamer/matchblanks.xsl"/>
<!-- define the output method for text -->
<xsl:output method="text"/>
<xsl:template match="/chapter[@id='1']">
  <!-- import the latex preamble and header -->
  <xsl:value-of select="$begin"/>
            <xsl:text>\subtitle{</xsl:text>
  <xsl:value-of select="$subtitle"/>
            <xsl:text>\\&#10;</xsl:text>
            <xsl:value-of select="chaptertitle"/>
  <xsl:text>}&#10;</xsl:text>
            <xsl:text>\begin{document}&#10;</xsl:text>
  <!--begin document -->
  <xsl:text>\frame{\titlepage} &#10;</xsl:text>
            <xsl:text>\section{Table of Contents}</xsl:text>
            <xsl:text>\begin{frame} &#10;</xsl:text>
            <xsl:text>\frametitle{Table of Contents} &#10;</x
  <xsl:text>\tableofcontents &#10;</xsl:text>
  <xsl:text>\end{frame} &#10;</xsl:text>
  <xsl:text>%Goals &#10;</xsl:text>
            <xsl:text>\section{Goals}</xsl:text>
            <xsl:text>\begin{frame} &#10;</xsl:text>
  <xsl:text>\frametitle{</xsl:text>
  <xsl:value-of select="chaptertitle"/>
  <xsl:text>}&#10;</xsl:text>
            <xsl:text>This lesson has the following goals:&#10
```

```
<chapter id="1">
        <chaptertitle>Working in the IT industry</chaptertitle>
        <chaptergoals>
                <goal>intrduce yourself and others</goal>
                <goal>ask an answer and personal questions</goal>
                <goal>talk about scheduled tasks</goal>
                <goal>use the alphabet and spell out workds</goal>
        </chaptergoals>
        <section>
                <sectionname>Meeting People</sectionname>
                <task module="gallery">
                        <taskname>Speaking</taskname>
                        <taskdescription>How do you greet people in your c
                        <illustration>
                                <image>
                                        <file>ch01f01a.png</file>
                                        <caption>image 1</caption>
                                </image>
                                <image>
                                        <file>ch01f01b.png</file>
                                        <caption>image 2</caption>
                                </image>
                                <image>
                                        <file>ch01f01c.png</file>
                                        <caption>image 3</caption>
```

(×) Variables — Global | Local

```
unknown version = 0.1
unknown title = English for Computer Technology
unknown subtitle = Practical Exercise Book
unknown extrainfo = Textbook for Vocational English
unknown author = Alaric Hamacher
unknown institution = Kwangwoon University
unknown begin = \documentclass[aspectratio=169,slidestop,compress
```

Stacks — Template | Input | Output | Mixed

```
→ chapter [chapter.xml]
xsl:template match="/chapter[@id='1']" [meta_to_beamer.xsl]
xsl:for-each [meta_to_beamer.xsl]
→ chaptergoals [chapter.xml]
xsl:for-each [meta_to_beamer.xsl]
→ goal [chapter.xml]
xsl:text [meta_to_beamer.xsl]
```

```
\date{\today}
%\version{0.1}
%\extrainfo{Textbook for Vocational English}
\logo{logo.png}
%\cover{cover.jpg}
\subtitle{Practical Exercise Book\\
Working in the IT industry}
\begin{document}
\frame{\titlepage}
\section{Table of Contents}\begin{frame}
\frametitle{Table of Contents}
\tableofcontents
\end{frame}
%Goals
\section{Goals}\begin{frame}
\frametitle{Working in the IT industry}
This lesson has the following goals:
\begin{itemize}
\item intrduce yourself and others
```

```
xsl:text [meta_to_beamer.xsl]
xsl:text [meta_to_beamer.xsl]
xsl:text [meta_to_beamer.xsl]
xsl:text [meta_to_beamer.xsl]
xsl:text [meta_to_beamer.xsl]
xsl:text [meta_to_beamer.xsl]
xsl:text [meta_to_beamer.xsl]
xsl:text [meta_to_beamer.xsl]
xsl:value-of [meta_to_beamer.xsl]
xsl:text [meta_to_beamer.xsl]
xsl:text [meta_to_beamer.xsl]
xsl:text [meta_to_beamer.xsl]
▼ xsl:for-each [meta_to_beamer.xsl]
  ▼ → chaptergoals [chapter.xml]
    ▼ xsl:for-each [meta_to_beamer.xsl]
```

Transformation Paused

meta_to_beamer.... | chapter.xml | Saxon (XSLT 2.0)

Fig. 6. XSLT processor xngr with debugger view

As result of these transformation new documents containing Latex source code for beamer and book document classes are produced. Compiling of these documents in Latex produces the desired output as textbook in pdf or graphical background for web casting and online courses.

5 Future Research

As proof of principle content production from a single metadata file was demonstrated. The versatility of XML stylesheet transformation bears a lot of more potential for economy of scale in content production: Further research could demonstrate how more different output targets can be achieved, such as interactive HTML quiz modules or graphical elements inside video presentations by creating shell scripts instead of Latex markup.

6 Conclusion

For many educators the additional work due to online courses in response to the Covid-19 pandemic has created the demand for new and efficient workflows. This paper presented an overview of various existing methods for creating teaching course contents. The research proposed a new workflow for authoring content on the basis of a single metafile. It demonstrated the principle how multiple publishing targets can be created from one single source and the potential for economies of scale.

Reducing the amount of work due to redundant authoring in incompatible software can give educators more time to focus on the teaching content. A multiple target publishing workflow as proposed in this research can help improving the quality of teaching from which teachers and students benefit.

7 Acknowledgement

The present research has been conducted by the Research Grant of Kwangwoon University in 2019.

8 References

[1] "Online learning goes mainstream as new normal," koreatimes, Apr. 23, 2020. http://www.koreatimes.co.kr/www/nation/2021/01/181_288366.html (accessed Jan. 16, 2021).

[2] H. Noprisson, "A Survey of the Online Learning Implementation During COVID-19 Outbreak," Int. J. Recent Contrib. Eng. Sci. IT, vol. 8, no. 4, p. 18, Dec. 2020, https://doi.org/10.3991/ijes.v8i4.17913

[3] R. M. Tawafak et al., "Impact of Technologies During COVID-19 Pandemic for Improving Behavior Intention to Use E-learning," International Journal of Interactive Mobile Technologies (iJIM), vol. 15, no. 01, Art. no. 01, Jan. 2021, Accessed: Jan. 13, 2021. [Online]. Available: https://online-journals.org/index.php/i-jim/article/view/17847. https://doi.org/10.3991/ijim.v15i01.17847

[4] S. A. Meo et al., “Comparison of the impact of PowerPoint and chalkboard in undergraduate medical teaching: an evidence-based study,” J Coll Physicians Surg Pak, vol. 23, no. 1, pp. 47–50, 2013.
[5] R. J. Craig and J. H. Amernic, “PowerPoint presentation technology and the dynamics of teaching,” Innovative higher education, vol. 31, no. 3, pp. 147–160, 2006. https://doi.org/10.1007/s10755-006-9017-5
[6] F. Crestani and M. Melucci, “A case study of automatic authoring: from a textbook to a hyper-textbook,” Data & knowledge engineering, vol. 27, no. 1, pp. 1–30, 1998. https://doi.org/10.1016/s0169-023x(97)00043-8
[7] G. Escobar and L. Kirsh, “Authoring tools for multimedia application development and network delivery,” Aug. 19, 1997.
[8] J. L. Pérez-Benedito, E. Q. Aragón, J. A. Alriols, and L. Medic, “Optical mark recognition in student continuous assessment,” IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 9, no. 4, pp. 133–138, 2014. https://doi.org/10.1109/rita.2014.2363005
[9] B. Inc, “Microsoft Word 2007 Mail Merge & Forms Quick Reference Guide,” 2007.
[10] M. J. Lu, E. A. Martinez, and S. Ranjan, “Mail merge integration techniques,” Feb. 25, 2014.
[11] S. A. Kazi, “A conceptual framework for web-based intelligent learning environments using SCORM-2004,” in IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings., 2004, pp. 12–15. https://doi.org/10.1109/icalt.2004.1357365
[12] X. Zhu, “Extending the SCORM Specification for references to the Open Content Object,” Educational Technology & Society, vol. 10, no. 1, pp. 248–264, 2007.
[13] A. Krewinkel and R. Winkler, “Formatting Open Science: agilely creating multiple document formats for academic manuscripts with Pandoc Scholar,” PeerJ computer science, vol. 3, p. e112, 2017. https://doi.org/10.7717/peerj-cs.112
[14] M. Urban, An introduction to LATEX. TEX users group, 1986.
[15] L. Lamport, LATEX: a document preparation system: user’s guide and reference manual. Addison-wesley, 1994.
[16] J. Duckett, HTML & CSS: design and build websites, vol. 15. Wiley Indianapolis, IN, 2011.
[17] J. N. Robbins, Learning web design: A beginner’s guide to HTML, CSS, JavaScript, and web graphics. O’Reilly Media, Inc., 2012. https://doi.org/10.5860/choice.50-6805
[18] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, “Extensible markup language (XML).,” World Wide Web Journal, vol. 2, no. 4, pp. 27–66, 1997.
[19] I. Williams, Beginning XSLT and XPath: Transforming XML documents and data. Wrox Press Ltd., 2009.
[20] D. R. Miller, K. S. Clarke, C. L. Hage, and K. S. Clarke, Putting XML to work in the library: tools for improving access and management. American Library Association, 2004.

9 Author

Alaric Hamacher, PhD in Computer Science and professor for 3D Contents and Virtual Reality in the Graduate School of Smart Convergence, Kwangwoon University. Alaric Hamacher graduated in directing and producing from the Academy for Television and Cinema Munich. His present research focusses on innovation, convergent publishing, 3D and VR.

Article submitted 2021-01-22. Resubmitted 2021-03-02. Final acceptance 2021-03-03. Final version published as submitted by the authors.