An Examination Question Paper Preparation System with Content-Style Separation and Bloom's Taxonomy Categorisation

Kah Ming Boon[1,2] and Lian Tze Lim[1]
[1] School of Engineering, Science and Technology, KDU College Penang,
32 Jalan Anson, Georgetown, Penang, Malaysia.
[2] Engineering and Environment, University of Northumbria at Newcastle,
Newcastle Upon Tyne, Tyne and Wear, United Kingdom.
boonkahming@hotmail.com, liantze@gmail.com

ABSTRACT

Most educators use WYSIWYG word processor applications to prepare examination question papers. Due to limitations of these software, however, educators often spend a disproportinate amount of time formatting their question papers to conform to institutional formatting guidelines, instead of focusing on the content and quality of the questions themselves. This project therefore intends to develop an examination question paper preparation system based on content-style separation principles, using XML, XSLT and LaTeX technologies. Using this system, educators will be able to set examination questions without having to consider their formatting. Instead, the system will take care of the formatting layout of the question paper with consistent adherance to given institutional style guides. In addition, the system will provide automatic detection of Bloom's Taxonomy level of each question, to aid educators in preparing examinations that are aligned to learning outcomes.

KEYWORDS

Examination question paper preparation, content-style separation, Bloom's Taxonomy, XML/XSLT, LaTeX

1 INTRODUCTION AND MOTIVATION

Education institutions typically have specific formatting guidelines for examination question papers to ensure style consistency and clarity. Most examiners prepare their question papers using WYSIWYG word processors, such as Microsoft Word. Unfortunately, a disproportional amount of time is often spent on formatting the question papers according to the institutional guidelines in such word processors, which could be better spent on improving the quality of the questions themselves. This is in part due to the lack of separation between style and content in word processors; and also because word processors are not purposed for professional typesetting. This project therefore intends to remove such unnecessary hurdles for the examiners, by using a semantic markup language, computer typesetting technology and natural language processing methods.

The system to be presented in this paper aims to help examiners in their question paper preparation and production by performing automatic formatting of the examination paper, thereby allowing examiners to concentrate on the content and quality of the questions themselves. The system will have three main features:

- A GUI front-end for creating and editing examination questions. (The user will not be concerned with the formatting and layout when defining questions.)
- A typesetting system that produces a correctly formatted examination question paper.
- Automatic categorisation of Bloom's Taxonomy level of each question.

The rest of this paper is organised as follows. Section 2 briefly reviews technologies that the examination question paper preparation system is built on, while section 3 explains the main components and processes of the system. Section 4 describes how the questions are categorised based on Bloom's Taxonomy levels. Some related work is mentioned in section 5, before concluding with suggestions of some future work in section 6.

2 LITERATURE REVIEW

This section briefly reviews principles and technologies forming the foundation of the examination question paper preparation system.

2.1 Content-Style Separation

What-You-See-Is-What-You-Get (WYSIWYG) word processors, such as Microsoft Word, enjoy great popularity because of their user-friendliness in creating and formatting text documents. However, this ease-of-use also leads to a serious drawback. The lack of separation between content and presentation style, coupled with

the user's unbound freedom to apply formattings, often leads to documents that do not conform to consistent formatting guidelines.

What is needed here is a hierarchical 'model of text' to organise information systematically, so that that information can be presented in a consistent manner. DeRose et al. [1] proposed that Standard Generalised Markup Language (SGML) [2] can provide such a hierarchical model. In practice, eXtensible Markup Language (XML) [3] (which is a subset of SGML) has been used to define markup languages of various document types and disciplines. Hackos [4, p.68] and Clark [5] support the use of XML in an organisation's information model, where XML is used to store data content separately. The XML file can then be transformed into some output format using Extensible Stylesheet Language Transformations (XSLT) [6]. This is achieved by the use of a transformation stylesheet, which contains declaratively defined templates that dictates how snippets of an XML file should be transformed into output text.

Another alternative for storing the examination paper data is by using a relational database. A relational database is a collection of data tables with relations between them. In terms of data storage, a relational database can outperform XML in certain aspects. Firstly, relational databases can handle and manage large volumes of data [7]. Besides that, by using normalisation procedures, data independence could be achieved more easily in a relational database than in an XML document tree structure.

Both XML and relational databases are excellent technologies for storing data and separating content from presentation. However, XML guarantees better portability [8], while relational database management systems from different vendors are not fully compatible. Besides, XML data is self-describing, and is able to model irregularities which might not be modelled easily using relational database [9]. For example, some questions in an examination might not contain any images, and XML can handle elements with heterogeneous structure. In this project, the examination questions stored by the system for each paper do not amount to any large data, and thus XML can handle the data storage well without the need of a relational database.

2.2 Output Generation and Typesetting

On the other hand, from the view of output generation, WYSIWYG word processors were not designed for typesetting. Due to the need for fast responsiveness, word processors do not usually incorporate complex typographical calculations (such as optimum hyphenations, line breaks and page breaks) when laying out text, and hence do not produce output of good typographic quality [10].

Conversely, the LaTeX document preparation system [11] produces PDF output of high typographic quality using the underlying TeX typesetting engine [12]. LaTeX is especially suitable for documents with complex structures and/or containing a lot of scientific or mathematical material. LaTeX is open source software and is available at no charge.

In addition, the LaTeX syntax is semi-semantic [13], which was designed to maximise the separation of content from presentation style by providing a logical markup interface rather than visual markup [14]. Authors should then be able to concentrate on the *content* (quality of questions) rather than the *style* (formatting) [15]. For example, the `exam.cls` LaTeX document class [16] allows authors to create examination questions using the following syntax:

```
\begin{questions}

\question[10]
Why is there air?

\question[15]
Why is the sky blue?

\end{questions}
```

with the following output:

1. (10 points) Why is there air?

2. (15 points) Why is the sky blue?

The LaTeX code above contains only the content and says nothing about the formatting, which can be customised separately in a style or class file. However, LaTeX has a steep learning curve, especially for users without any programming background [10, 13]. Instead of increasing productivity and effectiveness, requiring all examiners to interact directly with LaTeX to prepare question papers may prove to be a stumbling block instead. A GUI front-end for preparing the questions will therefore be provided to isolate LaTeX code away from examiners who prefer not to work with it.

2.3 Bloom's Taxonomy

Bloom's Taxonomy is an educational objectives classification system based on the student's understanding level essential for achievement [17]. Bloom's Taxonomy consists of six levels, with each level having a different degree of competency. These six levels are:

1. **Knowledge.** Students are required to remember information and recall what they have been learned

for questions on this level. Some of the question verbs include define, state, identify, list, label.

2. **Comprehension.** Questions on this level require students to be able to state the information in their own words, understand the translation and meaning of instructions, and interpret charts and graphs. Some of the question verbs include explain, summarize, interpret, convert, predict.
3. **Application.** Students are required to apply what they have learned to solve problems using suitable formula and algorithms.
4. **Analysis.** This level requires students to separate concepts into parts in order to understand the organizational structure and the relationship between parts. Some of the question verbs include analyze, compare, contrast, relate, differentiate.
5. **Synthesis.** Questions on this level allow students to invent a new structure from various elements and integrate parts together to form a complete system. Some of the question verbs include compile, create, design, construct, formulate.
6. **Evaluation.** Students are required to make judgments regarding the values of materials or ideas, method or solution to problems. Examples of question verb include defend, evaluate, justify, relate, support.

In this project, Bloom's Taxonomy will be used to analyse the expected competency levels of examination questions, so that different levels of questions could be prepared for students of different levels. Although there exists other learning objectives categorisation systems, Bloom's Taxonomy is selected as it can be easily applied to a range of question types due to its simple structure. Besides that, Bloom's Taxonomy is familiar to various academics and widely adopted [18].

2.4 Questions Analysis and Categorisation

A possible approach to categorising examination questions to appropriate Bloom's Taxonomy levels is by inspecting keywords. Important verbs and nouns are to be identified from the question text and mapped to appropriate levels in Bloom's Taxonomy. The level of the entire question can then be computed as a function of these mapped levels.

The advantages of using this method include faster execution and simpler algorithm implementation. This detection method however suffers from high inaccuracy, since questions are being categorised solely based on key nouns and verbs identified. An online test system was presented by Chang and Chung [19] to analyse and categorise the competency level of English questions based on Bloom's Taxonomy using verbs as keywords. A particular question would belong to a cognitive level of Bloom's Taxonomy when a matched keyword is found. However, only the 'Knowledge' level of Bloom's Taxonomy is efficiently categorised (75% correct matches) by the system using a keyword matching method. This method also suffers from ambiguities when a keyword may appear in multiple levels, which shall be discuss later.

Omar et al. [20] proposed an automated analysis of examination questions based on Bloom's Taxonomy using a rule-based approach. The proposed system also involved the application of natural language processing techniques to categorize exam questions, based on nouns and verbs identified. The system was intended to solve ambiguities, as keywords may be mapped to multiple cognitive levels [18].

Consider the questions below:

1. Write a complete Java program that takes 3 integers as inputs, and prints the largest integer among the 3 as output.
2. Write a statement to explain the choice made with proper justification.

Question 1 above requires the student to write a program in Java which performs certain tasks. The system might assume that this question belongs to the 'Knowledge' level in Bloom's Taxonomy because a matched keyword 'Write' is identified. The question actually belongs to the 'Synthesis' level of Bloom's Taxonomy as it requires the student to construct a program. Therefore, a conflict is raised as to which level the question should belong to. Similarly in Question 2, the phrase 'Write a statement' could possibly be of 'Knowledge' or other cognitive levels, depending on the exact nature of the 'statement'. The next keyword 'explain' suggests that the 'statement' should give an explanation on the choice made, which will belong to the 'Comprehension' level. However, reading to the end of the question, it is revealed that a justification should be made on the choice made, thus making the question an 'Evaluation' one. Omar et al. [20] introduced 'category weighting' to solve this problem. Conflicting categories are assigned different weights, which are calculated by subject matter experts. However, this method requires a training set of questions to obtain the optimised weights for all possible question patterns.

3 SYSTEM DESIGN AND PROCESSES

A examination paper preparation system prototype is currently in development using the C#.Net language. The system is built on XML, XSLT and LaTeX, which were reviewed in earlier sections. Figure 1 shows a high-level overview of the system's design and processes.

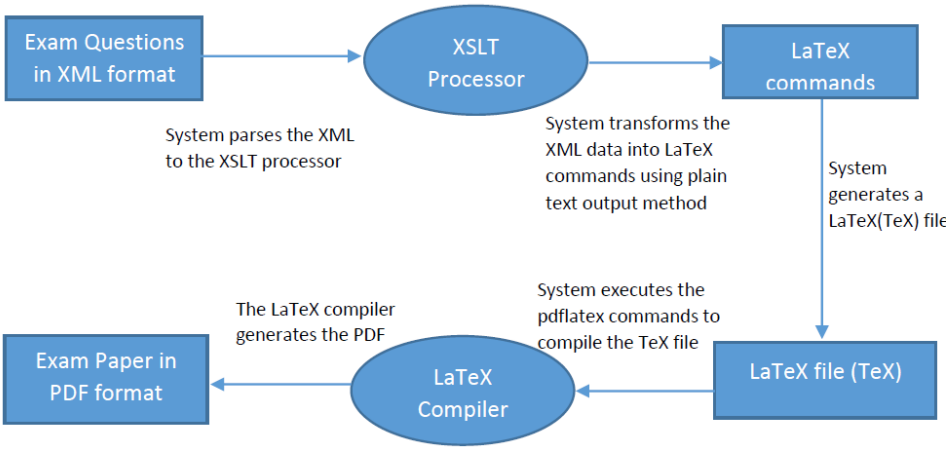


Figure 1: Overview of system design and processes

The examiner first creates an examination paper and adds questions, using a GUI front-end. The question paper will be saved to disk as an XML file. The XML file is then transformed using an XSLT transformation stylesheet into a LaTeX file. The LaTeX file is then compiled in conjunction with a LaTeX class implementing an institutional's style guide. The final output is a fully formatted PDF file.

The following sections will discuss each stage of the system in more detail. The system can be customised for different institutions by modifying the LaTeX document class to provide for different layouts and formatting (see section 3.4), and the Bloom's Taxonomy levels keyword mapping list (see section 4).

3.1 Examination Paper Document Model

An XML document model has been designed in order to cater for the following requirements in a college- or university-level examination question paper:

- The following information about an examination question paper should be recorded:
  - name and/or campus of institution;
  - name and of faculty;
  - name and/or code of programme;
  - name and/or code of subject;
  - semester or date of examination;
  - duration of examination;
  - name of examiner.
- A question paper may contain multiple sections of questions.
- Questions could be either of multiple-choice or open-ended.
- Each question should have a mark allocation.
- All multiple-choice questions in a section should be allocated equal marks.
- Questions could contain parts, with each having its own mark allocation.
- Question texts may contain simple text formatting, images, code snippets (for programming subjects).
- Questions may contain LaTeX syntax to create special material, e.g. mathematical equations.

The examination questions are saved to an XML file on the hard disk. Snippets from an examination question paper XML file can be found in Appendix A.

3.2 GUI Front-End

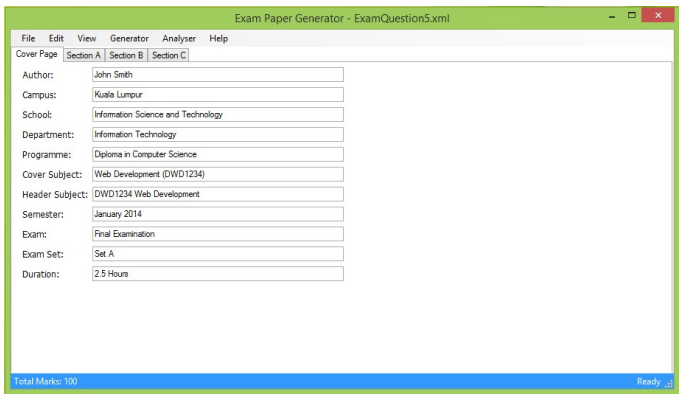
A GUI front-end, as shown in Figure 2, has been developed for creating and editing an examination question paper.

Information about the examination is collected (Figure 2(a)) to generate the cover page, page headers and footers (if any). The examiner may then add any number of question sections, which may contain multiple choice questions (Figure 2(b)) or open-ended questions (Figure 2(c)). Each open-ended question may consist of parts. Examiners may also specify the mark allocations. Each question may also contain images and code listings (for computer programming subjects). Examiners familiar with LaTeX may also use LaTeX syntax directly in the editing areas, e.g. to add mathematical material.

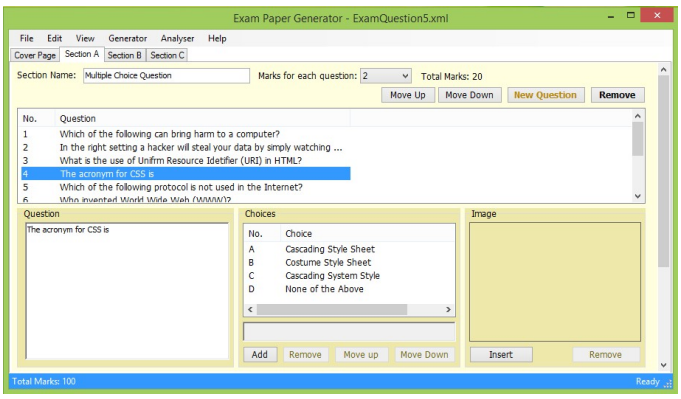
As shown in the screenshots, the system will display the running total marks for each question section, as well as for the entire paper. This would help examiners keep track of the overall mark allocations. Examiners will also be able to import questions from other existing examination questions XML files.

3.3 XSLT Transformation Stylesheet

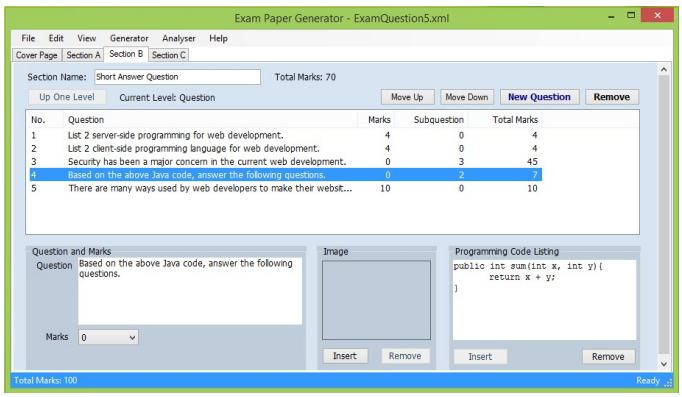
An XSLT stylesheet is created to transform the examination XML file to a LaTeX file. The stylesheet contains transformation templates, which function as rules for transforming matching XML blocks to output text.

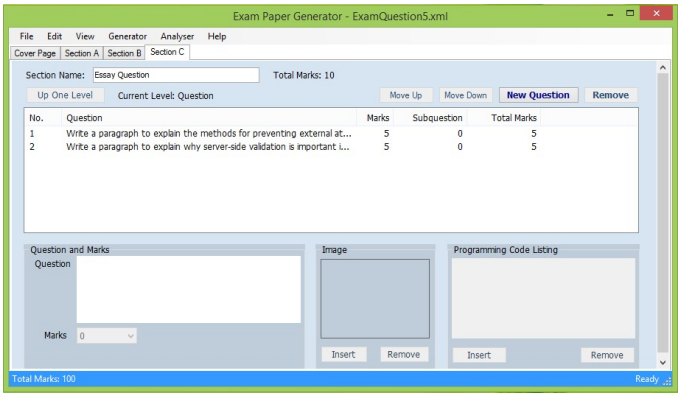


(a) Information of the examination



(b) Adding multiple choice questions





(c) Adding open-ended questions and mark allocations. Options of including images and program code listings are provided.

Figure 2: Screenshots of the examination preparation tool GUI

The XSLT language contains many powerful constructs for more sophisticated matching, processing and transformation. The following example shows only a basic usage. This XSLT template would match `<question>...</question>` blocks in the XML file, to produce the necessary LaTeX code.

```
<xsl:template match="question">
  \Q[<xsl:value-of select="@marks" />]
  <xsl:value-of select="string"/>
</xsl:template>
```

XSLT template for `question`

```
<question marks="4">
  <string>List 2 server-side programming for
      web development.</string>
</question>
```

Matching XML `question` block

```
\Q[4] List 2 server-side programming for web
    development.
```

LaTeX code as transformation output

Figure 3: XSLT processing for LaTeX code generation

A more complete example generated LaTeX examination file can be found in Appendix B.

3.4 Customised `exam` LaTeX Class File

To implement formatting and layouts required by individual institutions, a customised LaTeX class file can be created based on the existing `exam.cls` class [16]. The new class file, e.g. `ACME-exam.cls`, can contain formatting directives for different elements of the examination paper including fonts, headers, footers, alignment of the cover page, section headings, questions and mark allocations.

LaTeX files generated by the XSLT processor will load the customised class file via the document declaration `\documentclass{ACME-exam}`. The examination paper preparation system will then invoke the `pdflatex` engine to process the LaTeX file resulting from the XSLT transformation, to produce the final PDF file of the examination question paper. The LaTeX class file can also be used to create examination question papers using LaTeX syntax directly by examiners who are familiar with the language.

4 QUESTION ANALYSIS FOR BLOOM'S TAXONOMY CATEGORISATION

Natural language procesing techniques are used to analyse and categorise examination questions according to

Bloom's Taxonomy congnitive levels. We currently employ a very basic algorithm as follows:

STEP 1. A list of action verbs and related keywords, together with their mappings to Bloom's Taxonomy levels, is first prepared. Some sample entries from the mapping file are shown below:

> **Level 1 (Knowledge)**
> list, define, definition, label, identify...
>
> **Level 2 (Comprehension)**
> explain, explanation, rewrite, summarise...
> ...

STEP 2. For every question, each word in the question text is first tagged with its part-of-speech (POS) using the Stanford POS-Tagger [21]. For example, the question text

> Write a short paragraph to explain why server-side validation is important in a website.

would be POS-tagged thus:

> Write/VB a/DT short/JJ paragraph/NN to/TO explain/VB why/WRB server-side/JJ validation/NN is/VBZ important/JJ in/IN a/DT website/NN ./.

In the above result, the POS tag (from the Penn Treebank tagset [22]) of each word is shown after the slash character. For example, 'write' is a verb in its base form (VB); 'a' is a determiner (DT); 'short' is and adjective (JJ), 'paragraph' is a singular common noun (NN).

STEP 3. Extract phrases and clauses of interest. This is done by regular expression matching of the POS sequences, e.g.

```
VB:*:NN
VB:*:NN:TO:VB
```

where the pattern * can match any sequence. The second pattern would match the extracted phrase

> Write/VB a/DT short/JJ paragraph/NN to/TO explain/VB

from the previous example.

STEP 4. Map verbs and nouns in the extracted verb phrases to their respective Bloom's Taxonomy levels. The highest taxonomy level is assigned to the question. In the previous example, the only word that appears in the keyword-mapping list (from Step 1) is 'explain', which is mapped to the 'Comprehension' level. The algorithm thus assigns 'Comprehension' as the category of this example question.

This function is also made available in the GUI front-end (Figure 4). As the current algorithm is quite simple, the categorisation may not always be accurate, but rather serves as a reference estimation to the examiner.

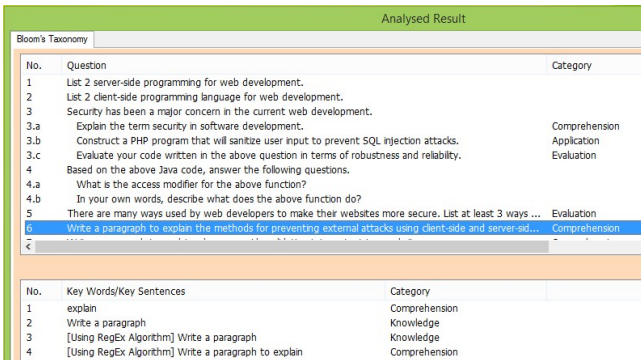


Figure 4: Categorising questions to Bloom's Taxonomy levels using the GUI front-end.

5 RELATED WORK

A number of educational institutions adopt the use of LaTeX for creating examination question papers. Some of these institutions creating their own LaTeX class files for doing so, including the University of Oxford [23], the Australian National University [24], and the Victoria University of Wellington [25]. To use these class files, examiners are expected to author the examination papers in LaTeX directly. Derani [26] created a Web-based system for submitting examination paper LaTeX files, but no tools are provided to aid examiners in producing the LaTeX files.

Cen et al. [27] created a system that generates question papers from a question bank based on user-defined assessment level settings. Their system generates Microsoft Word documents, and places no emphasis on formatting details. Bridgeman et al. [28] created a LaTeX package which, besides generating assignment papers, also provides tools for instructors to generate differing versions of computer science problems by using the idea of parameterising them.

6 CONCLUSION AND FUTURE WORK

An examination question paper preparation system based on content-style separation principles has been presented. The aim of the system is to relieve examiners from dealing with visual and stylistic formatting details, so that they may concentrate on preparing examination questions of good quality. The system, developed using the C#.Net language, uses an XML document model to store information about the examination questions, which will be transformed into a LaTeX file using XSLT transformation. The final form of the examination question paper is a PDF file, using a custom-written LaTeX

document class which contains styling directives adhering to institutional formatting guidelines. A GUI front-end is provided to facilitate the creation and editing of the examination questions.

The examination question paper preparation system is currently a work in progress. Some future features may include:

- Allows creation of answer scheme using GUI front-end. (The LaTeX document class for this is already available.)
- Improved text editor to aid insertion of tables, mathematical material, bullet lists.
- Improved Bloom's Taxonomy categorisation algorithm. Possible enhancements include making use of the phrase structure, machine learning approach, or other language resources.

We will soon make the system available for internal testing by interested lecturers, with the hope that this will aid them in their examination preparation efforts.

REFERENCES

[1] S. J. DeRose, D. G. Durand, E. Mylonas, and A. H. Renear, "What is text, really?" *SIGDOC Asterisk J. Comput. Doc.*, vol. 21, no. 3, pp. 1–24, Aug. 1997.

[2] M. Bryan, *SGML*. Addison-Wesley, 1988.

[3] 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.

[4] J. T. Hackos, *Content management for dynamic Web delivery*. John Wiley & Sons, Inc., 2002.

[5] D. Clark, "Content management and the separation of presentation and content," *Technical communication quarterly*, vol. 17, no. 1, pp. 35–60, 2007.

[6] D. Tidwell, *XSLT*. O'Reilly, 2001.

[7] J. Robie and D. Bartels, "A comparison between relational and object oriented databases for object oriented application development," *POET Software Corporation*, pp. 800–950, 1994.

[8] A. Mengel and W. Lezius, "An XML-based representation format for syntactically annotated corpora," in *Proceedings of LREC 2000*, Athens, Greece, 2000.

[9] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu, "A query language for xml," *Computer networks*, vol. 31, no. 11, pp. 1155–1169, 1999.

[10] K. L. Warnock, "Internet document services," Oct. 2 2007, uS Patent 7,277,925.

[11] L. Lamport, *LaTeX: A Document Preparation System: User's Guide and Reference Manual*, 2nd ed. Reading, MA, USA: Addison-Wesley, 1994.

[12] D. E. Knuth, *The TeXbook*, ser. Computers and Typesetting. Reading, MA, USA: Addison-Wesley, 1986, vol. A.

[13] M. Kohlhase, "Using LaTeX as a semantic markup format," *Mathematics in Computer Science*, vol. 2, no. 2, pp. 279–304, 2008.

[14] F. Mittelbach and C. Rowley, "The LaTeX3 project," *MAPS*, vol. 93, pp. 95–99, 1993.

[15] S. D. Webb, "Accessibility of university mathematics," *MSOR Connections*, vol. 11, no. 1, pp. 42–47, 2011.

[16] P. S. Hirschhorn, "`exam` – package for typesetting exam scripts," 2011, last accessed: February 23, 2014. [Online]. Available: http://ctan.org/tex-archive/macros/latex/contrib/exam/

[17] B. S. Bloom, M. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, "Taxonomy of educational objectives: Handbook i: Cognitive domain," *New York: David McKay*, vol. 19, p. 56, 1956.

[18] K. O. Jones, J. Harland, J. M. Reid, and R. Bartlett, "Relationship between examination questions and Bloom's taxonomy," in *Frontiers in Education Conference, 2009. FIE'09. 39th IEEE*. IEEE, 2009, pp. 1–6.

[19] W.-C. Chang and M.-S. Chung, "Automatic applying bloom's taxonomy to classify and analysis the cognition level of english question items," in *Proceedings of 2009 Joing Conferences on Pervasive Computing (JCPC)*. IEEE, 2009, pp. 727–734.

[20] N. Omar, S. S. Haris, R. Hassan, H. Arshad, M. Rahmat, N. F. A. Zainal, and R. Zulkifli, "Automated analysis of exam questions according to bloom's taxonomy," *Procedia-Social and Behavioral Sciences*, vol. 59, pp. 297–303, 2012.

[21] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 173–180.

[22] A. Taylor, M. Marcus, and B. Santorini, "The penn treebank: an overview," in *Treebanks*. Springer, 2003, pp. 5–22.

[23] H. Lowe, "LaTeX style and template files used for examination papers," 2013, last accesssed: February 23, 2014. [Online]. Available: http://www.maths.ox.ac.uk/teaching-staff/examinations/exam-sty-files

[24] T. Kam, "`ANUfinalexam` – LaTeX document shell for anu final exam," 2009, last accesssed: February 23, 2014. [Online]. Available: http://www.ctan.org/pkg/anufinalexam

[25] M. Ray, "Exam templates," 2013, last accessed: February 23, 2014. [Online]. Available: http://ecs.victoria.ac.nz/Support/ExamTemplates

[26] S. A. Derani, "LaTeX style files for uitm exam papers and online submission," Bachelor Degree Thesis, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 2007.

[27] G. Cen, Y. Dong, W. Gao, L. Yu, S. See, Q. Wang, Y. Yang, and H. Jiang, "A implementation of an automatic examination paper generation system," *Mathematical and Computer Modelling*, vol. 51, no. 11, pp. 1339–1342, 2010.

[28] S. Bridgeman, M. T. Goodrich, S. G. Kobourov, and R. Tamassia, "Sail: a system for generating, archiving, and retrieving specialized assignments using LaTeX," in *ACM SIGCSE Bulletin*, vol. 32, no. 1. ACM, 2000, pp. 300–304.

A EXAMINATION PAPER XML FILE

```
<?xml version="1.0" encoding="utf-8"?>
<examPaper>
  <preAttribute>
    <author>John Smith</author>
    <campus>Main</campus>
    <school>School of Engineering and Information
        Sciences</school>
    <dept>Information Technology Department</dept>
    <programme>Diploma in Computer Science</
        programme>
    <coversubj>Web Development (DWD1234)</coversubj>
    <headersubj>DWD1234 Web Development</headersubj>
    <semester>January 2014</semester>
    <exam>Final Examination</exam>
    <examset>Set A</examset>
    <duration>2.5 Hours</duration>
  </preAttribute>
  <sections>
    <section type="mcq" eachMarks="2" name="Multiple
        Choice Question">
      <question>
        <string>Which of the following can bring
            harm to a computer?</string>
        <choices>
          <choice>Vulnerability</choice>
          <choice>phish</choice>
          <choice>threat</choice>
          <choice>spoof</choice>
        </choices>
      </question>...
    </section>
    <section type="sq" name="Short Answer Question">
      <question marks="4">
        <string>List 2 server-side programming for
            web development.</string>
      </question>
      <question marks="4">
        <string>List 2 client-side programming
            language for web development.</string>
      </question>
      <question marks="0">
        <string>Security has been a major concern in
            the current web development.</string>
        <subquestion>
          <question marks="10">
            <string>Explain the term security in
                software development.</string>
          </question>
          <question marks="20">
            <string>Construct a PHP program that will
                sanitize user input to prevent SQL
                injection attacks.</string>
          </question>
          <question marks="15">
            <string>Evaluate your code written in the
                above question in terms of
                robustness and reliability.</string>
          </question>
        </subquestion>
      </question> ...
    </section>
  </sections>
</examPaper>
```

B EXAMPLE GENERATED LaTeX FILE

```
\documentclass{ACME-exam}

\author{John Smith}
\campus{Main}
\school{School of Engineering and Information
    Sciences}
\dept{Information Technology Department}
\programme{Diploma in Computer Science}
\coversubj{Web Development (DWD1234)}
\headersubj{DWD1234 Web Development}
\semester{January 2014}
\exam{Final Examination}
\examset{Set A}
\duration{2.5 Hours}

\begin{document}
\instruction{Answer ALL questions.}

\begin{exampaper}

\examsection{Multiple Choice Questions.}
\begin{Questions}
\Q[2]
Which of the following can bring harm to a
    computer?
\begin{choices}
\choice Vulnerability
\choice phish
\choice threat
\choice spoof
\end{choices}
...

\examsection{Short Answer Question}
\begin{Questions}
\Q[4] List 2 server-side programming for web
    development.

\Q[4] List 2 client-side programming language
    for web development.

\Q Security has been a major concern in the
    current web development.

\begin{Parts}
\Qa[10] Explain the term security in software
    development.
\Qa[20] Construct a PHP program that will
    sanitize user input to prevent SQL
    injection attacks.
\Qa[15] Evaluate your code written in the
    above question in terms of robustness and
    reliability.
\end{Parts}
...
\end{exampaper}
\end{document}
```

C EXAMPLE FINAL PDF

ACME UNIVERSITY

MAIN CAMPUS

SCHOOL OF ENGINEERING AND INFORMATION SCIENCES

INFORMATION TECHNOLOGY DEPARTMENT

DIPLOMA IN COMPUTER SCIENCE

WEB DEVELOPMENT (DWD1234)

FINAL EXAMINATION
JANUARY 2014 SEMESTER

DURATION: 2.5 HOURS

TOTAL MARKS: 100

General Instructions:

1. All answers are to be written in black or blue ink on the answer booklet provided.
2. This examination consists of **THREE** sections – Sections A, B and C. Answer **ALL** questions in **ALL** sections.
3. This examination paper consists of **TWO** pages of questions.
4. Students caught copying, or having unauthorized printed materials, pieces of written materials, or any form of action with intention to cheat and/or copy will be penalized.

Answer ALL questions.

Section A: Multiple Choice Questions.

1. Which of the following can bring harm to a computer?
   A. Vulnerability
   B. phish
   C. threat
   D. spoof

2. In the right setting a hacker will steal your data by simply watching what you type.
   A. snagging
   B. spying
   C. social engineering
   D. shoulder surfing

3. What is the use of Unifrm Resource Idetifier (URI) in HTML?
   A. To create a frame document.
   B. To create an image map in the webpage.
   C. To customize the image in the webpage.
   D. To identify a name or a resource on the Internet.

4. The acronym for CSS is
   A. Cascading Style Sheet
   B. Costume Style Sheet
   C. Cascading System Style
   D. None of the Above

5. Which of the following protocol is not used in the Internet?
   A. Telnet
   B. WIRL
   C. HTTP
   D. Gopher

6. Who invented World Wide Web (WWW)?
   A. Blaise Pascal
   B. Charles Babbage
   C. Herman Hollerith
   D. Tim Berners-Lee

7. Forms in HTML is used
   A. to display a web page within a web page.
   B. to display animation effect.
   C. to collect user's input
   D. None of the Above.

8. Iframe in HTML is used
   A. to display contents of email.

   B. to display a web page with animation effect.
   C. to collect user's input
   D. All of the Above.

9. The acronym for FTP is
   A. File Transaction Protocol
   B. File Transmission Protocol
   C. File Translation Protocol
   D. File Transfer Protocol

10. Which of the following is consiered as Web Servie Platform Elements?
    A. UDDI
    B. WSDL
    C. SOAP
    D. All of the Above

[20 marks]

Section B: Short Answer Question

1. List 2 server-side programming for web development. (4 marks)
2. List 2 client-side programming language for web development. (4 marks)
3. Security has been a major concern in the current web development.
   a) Explain the term security in software development. (10 marks)
   b) Construct a PHP program that will sanitize user input to prevent SQL injection attacks. (20 marks)
   c) Evaluate your code written in the above question in terms of robustness and reliability. (15 marks)

```
public int sum(int x, int y){
   return x + y;
}
```

4. Based on the above Java code, answer the following questions.
   a) What is the access modifier for the above function? (2 marks)
   b) In your own words, describe what does the above function do? (5 marks)
5. There are many ways used by web developers to make their websites more secure. List at least 3 ways to increase a website security. Justify your answer. (10 marks)

[70 marks]

Section C: Essay Question

1. Write a paragraph to explain the methods for preventing external attacks using client-side and server-side validations. (5 marks)
2. Write a paragraph to explain why server-side validation is important in a website. (5 marks)

[10 marks]

THE END

Prepared by John Smith
Information Technology Department
School of Engineering and Information Sciences