Free and Open Source Software for Technical Texts Editing, Its Advantages and Experience of Usage on TMM Training in Bauman University

Alexander Titov(✉) and Andrei Vukolov

Aerospace Faculty, Bauman Moscow State Technical University, Moscow, Russian Federation
aserwqq@gmail.com, twdragon@bmstu.ru

Abstract. Creation of the documentation is practically hardest part of any engineering, software development or scientific project containing large amount of knowledge. In Bauman University every TMM project includes main document (thesis) which contains explanation of all calculations performing through development of the project. To reduce time of preparation for such technical texts the complete software solution with formulae support needed. In this paper usage of LaTeX (as common free solution) is considered in comparison with most known free and proprietary solutions such as MS Word, Open Office and T-Flex DOCs.

Keywords: Open source · Free software · TMM
Engineering education · Text documentation · Technical texts
LaTeX · MS Word · Libre Office · Open Office · CAD/CAM

1 Introduction

Technical progress made computer text editing solutions the only useful instruments for technical text preparation in engineering education. However, most solutions in this area are proprietary. Such products are very difficult to use in permanently changing environment, for example in shared computer classes. They require licensing control services that are often difficult to support through many machines. Also when educational organization owns more than 3–4 different proprietary solutions it is always big problem to keep classes ready. It happens because license control solutions from different vendors consume large amount of computational resources and often require newest versions of operating systems. The special problem is incorrect inheritance of license requirements. It can make it impossible to deploy documentation and/or instruments developed by student onto his own machine or the machine placed outside main university's network.

J. C. García-Prada and C. Castejón (Eds.): *New Trends in Educational Activity in the Field of Mechanism and Machine Theory*, MMS 64, pp. 208–215, 2019.
https://doi.org/10.1007/978-3-030-00108-7_23

To avoid described problems and constraints many open-source communities such as GNU project developed special documents called *Free Licenses* [1,2,5,6]. These documents allows user who use the software issued under free licenses, to develop and distribute his own programs, documentation and solutions based on, and place texts, code and/or executables into public domain with well-known requirements easy for understanding [19]. Clearing and granting function for permission of usage of free licenses is carried by Free Software Foundation (FSF) and Open Source Initiative (OSI) organizations. Now usage of free software grows active through all over the world [14] including Bauman University [9,23]. Besides free licenses the engineers can distribute products, graphics and documentation freely without permission of redistribution.

The most common types of texts developed in Bauman University are specification, development thesis and datasheet. Preparation of each type of document is an important part of educational process. In typical environment it requires several specialized software solutions. In Bauman University the specifications are usually prepared using the same software than the graphical documentation [9,23,24]. Development thesis and datasheets are usually prepared using common office solutions. On this stage, however, it is important to disconnect the process with concrete solution because the student goes to achieve possibility to *develop a system*, not to *use a program*.

2 Peculiarities of Text Documentation Developed in Bauman University

Text documents prepared by students in Bauman University contain bibliography, large amount of graphics and formulae (Fig. 1). For ordinary project there are no special requirements on software. But the main solution which are in active usage on almost all faculties and departments is *Microsoft Word*. It is proprietary solution which can not be used outside university's network. Some documents also require special fonts and typesetting such as ISO or GOST font kits. These typesetting kits are also proprietary and provided by such vendors as *Autodesk* and *TopSystems*. The technical texts in Bauman University often

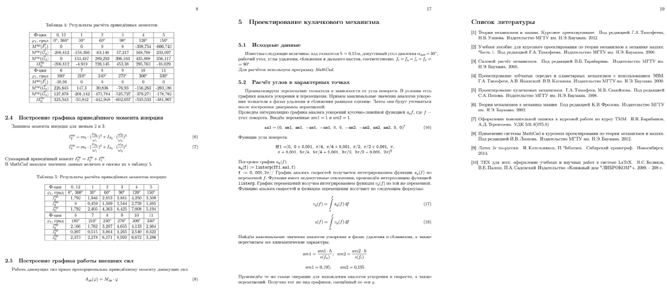
8

Таблица 4: Результаты расчёта приведённых моментов

Ф-ция	0, 12	1	2	3	4	5
φ_1, град	0°, 360°	30°	60°	90°	120°	150°
$M^{пр}(\vec{F}_c)$	0	0	0	0	-308,754	-606,743
$M^{пр}(\vec{G}_2)$	-206,812	-158,366	-63,148	57,217	168,708	233,997
$M^{пр}(\vec{G}_3)$	0	153,447	289,293	396,163	435,808	356,117
$M^{пр}_{\Sigma}$	-206,812	-4,919	226,145	453,38	295,763	-16,629
Ф-ция	6	7	8	9	10	11
φ_1, град	180°	210°	240°	270°	300°	330°
$M^{пр}(\vec{F}_c)$	-30,98	0	0	0	0	0
$M^{пр}(\vec{G}_2)$	226,645	147,3	30,836	-76,93	-156,263	-203,186
$M^{пр}(\vec{G}_3)$	127,878	-201,142	-473,784	-525,727	-379,271	-178,782
$M^{пр}_{\Sigma}$	323,543	-53,842	-442,948	-602,657	-535,533	-381,967

2.4 Построение графика приведённого момента инерции

Запишем моменты инерции для звеньев 2 и 3:

$$I_2^{пр} = m_2 \cdot (\frac{v_{S_2}}{\omega_1})^2 + I_{S_2} \cdot (\frac{\omega_2}{\omega_1})^2 \quad (6)$$

$$I_3^{пр} = m_3 \cdot (\frac{v_{S_3}}{\omega_1})^2 + I_{S_3} \cdot (\frac{\omega_3}{\omega_1})^2 \quad (7)$$

Суммарный приведённый момент $I_{\Sigma}^{пр} = I_2^{пр} + I_3^{пр}$.
В MathCad находим значения данных величин и сносим их в таблицу 5.

Таблица 5: Результаты расчёта приведённых моментов инерции

Ф-ция	0, 12	1	2	3	4	5
φ_1, град	0°, 360°	30°	60°	90°	120°	150°
$I_2^{пр}$	1,792	1,946	2,853	3,881	4,250	3,509
$I_3^{пр}$	0	0,459	1,509	2,544	2,759	1,685
$I_{\Sigma}^{пр}$	1,792	2,405	4,363	6,425	7,008	5,194
Ф-ция	6	7	8	9	10	11
φ_1, град	180°	210°	240°	270°	300°	330°
$I_2^{пр}$	2,166	1,762	3,207	4,655	4,133	2,664
$I_3^{пр}$	0,207	0,515	3,064	4,265	2,540	0,622
$I_{\Sigma}^{пр}$	2,373	2,278	6,271	8,920	6,672	3,286

2.5 Построение графика работы внешних сил

Работа движущих сил прямо пропорциональна приведённому моменту движущих сил:

$$A_{дв}(\varphi) = M_{дв} \cdot \varphi \quad (8)$$

17

5 Проектирование кулачкового механизма

5.1 Исходные данные

Известны следующие величины: ход толкателя $h = 0{,}15$ м, допустимый угол давления $\alpha_{доп} = 30°$, рабочий угол, углы удаления, сближения и дальнего выстоя, соответственно, $f_r = f_u = f_c = f_d = 90°$.
Для расчётов используем программу *MathCad*.

5.2 Расчёт углов в характерных точках

Проанализируем перемещение толкателя в зависимости от угла поворота. В условии есть графики аналога ускорения и перемещения. Примем максимальные значения аналогов ускорения толкателя в фазах удаления и сближения равными единице. Затем они будут уточняться после построения диаграммы перемещений.
Проведём интерполяцию графика аналога ускорений кусочно-линейной функцией $a_q f$, где f — угол поворота. Введём переменные $am1 = 1$ и $am2 = 1$.

$$\mathtt{aa1} = (0,\ \mathtt{am1},\ \mathtt{am1},\ -\mathtt{am1},\ -\mathtt{am1},\ 0,\ 0,\ -\mathtt{am2},\ -\mathtt{am2},\ \mathtt{am2},\ \mathtt{am2},\ 0,\ 0)^T \quad (16)$$

Функция угла поворота:

$$\mathtt{ff1} = (0,\ 0+0{,}001,\ \pi/4,\ \pi/4+0{,}001,\ \pi/2,\ \pi/2+0{,}001,\ \pi,\ \pi+0{,}001,\ 5\pi/4,\ 5\pi/4+0{,}001,\ 3\pi/2,\ 3\pi/2+0{,}001,\ 2\pi)^T$$

Построим график $a_q(f)$:
$\mathtt{a_q(f) := linterp(ff1, aa1, f)}$
$\mathtt{f := 0, 001, 2\pi}$ // График аналога скоростей получается интегрированием функции $a_q(f)$ по переменной f. Функция имеет недопустимые отклонения, произведём интерполяцию функцией `linterp`. График перемещений получим интегрированием функции $v_q(f)$ по той же переменной. Функцию аналога скоростей и функцию перемещения получают по следующим формулам:

$$v_q(f) = \int_0^f a_q(f)\,df \quad (17)$$

$$s(f) = \int_0^f v_q(f)\,df \quad (18)$$

Найдём максимальные значения аналогов ускорения в фазах удаления и сближения, а также пересчитаем все кинематические параметры.

$$am1 = \frac{am1 \cdot h}{s(f_u)}; \quad am2 = \frac{am2 \cdot h}{s(f_r)}$$

$$am1 = 0{,}195; \quad am2 = 0{,}195$$

Произведём те же самые операции для нахождения аналогов ускорения и скорости, а также перемещений. Получим тот же вид графиков, смещённый по оси y.

19

Список литературы

[1] Теория механизмов и машин. Курсовое проектирование. Под редакцией Г.А. Тимофеева, Н.В. Умнова. Издательство МГТУ им. Н.Э. Баумана. 2012

[2] Учебное пособие для курсового проектирования по теории механизмов и механике машин. Часть 1. Под редакцией Г.А. Тимофеева. Издательство МГТУ им. Н.Э. Баумана. 2000.

[3] Силовой расчёт механизмов. Под редакцией В.Б. Тарабарина. Издательство МГТУ им. Н.Э. Баумана. 2000.

[4] Проектирование зубчатых передач и планетарных механизмов с использованием ЭВМ. Г.А. Тимофеев, А.В. Яминский. В.В. Каганова. Издательство МГТУ им. Н.Э. Баумана. 2000.

[5] Проектирование кулачковых механизмов. Г.А. Тимофеев, М.В. Самойлова. Под редакцией С.А. Попова. Издательство МГТУ им. Н.Э. Баумана. 1998.

[6] Теория механизмов и механика машин. Под редакцией К.В. Фролова. Издательство МГТУ им. Н.Э. Баумана. 2002.

[7] Оформление пояснительной записки к курсовой работе по курсу ТММ. Н.Н. Барабашов, А.Д. Терентьева. УДК 531.8(075.8)

[8] Применение системы MathCad в курсовом проектировании по теории механизмов и машин. Под редакцией И.В. Леонова. Издательство МГТУ им. Н.Э. Баумана. 2012.

[9] Латех 2ε по-русски. И. Котельников, П. Чеботаев. Сибирский хронограф. Новосибирск. 2014.

[10] ТЕХ для всех: оформление учебных и научных работ в системе LaTeX. Н.С. Беляков, В.Е. Палош, П.А. Садовский Издательство «Книжный дом "ЛИБРОКОМ"». 2009. - 208 с.

Fig. 1. Typical development thesis pages

required to be prepared in PDF format. Windows environment does not provide any native solution for conversion from *Word* native DOC format to PDF instead of proprietary *Adobe Acrobat*. This possibility is provided in open source office suite named *LibreOffice* with some restrictions: for example, it is not possible to embed multimedia content into PDF file using free software. After described pipeline the developed document is strictly connected with solution: the license places restrictions on any case of its usage. Possibility of the free redistribution of developed documents can be obtained only with free pipelines.

3 Advantages of Free Software in Training Process

Here and below the advantages of free software in comparison with proprietary solutions will be given for the following list of features:

1. Cross-platform interoperability;
2. Processing of formulae;
3. Processing of raster graphics;
4. Generating and processing of vector(ized) graphics.

3.1 Cross-platform Interoperability

The first step of text document preparation is the page and exterior typesetting. WYSIWYG[1] solutions like *Word* or *LibreOffice* provide special dialogs This type of document type setting requires to repeat configuration for each new document which are being prepared using the solution. To avoid this some WYSIWYG solutions, including *Word* provide template mechanism which allows user to create several documents inherited from single special prepared one called *document template* (.DOT-file). This model makes document dependent on the solution and even on its version because the templates prepared within different versions of the same program would be uncompatible due to embedded VBA core.

Unlike this, LaTeX does not follow WYSIWYG methodology. It provides TeX programming language for type setting and markup configuration tasks. In this case the document consists of plain text only and it has special part called *preamble* which holds the same functions as the template: to describe minor changes of the exterior that aren't inherited:

```
\documentclass[graybox,stagethree,,online]{svmult}
\usepackage{amsmath,amssymb,url,listings}
\usepackage{epsfig,epstopdf}
\makeindex
\graphicspath{{img/}}
```

All other type setting instructions are encapsulated within the special *style file*. It is not a template but the processing rules which are strictly prescribed by LaTeX language standard. This variant is truly cross-platform because all files as

[1] **WYSIWYG** [3] ("What You See Is What You Get")—document preparation methodology which prescribes identical view of the document on editor's screen and within any other type of view.

consist of the plain text as document itself [4]. The same set of source files produce also the same result on all operating systems and hardware platforms which LaTeX processing engine had been realized on. Also LaTeX processing engine may produce output in both PostScript, PDF or DVI formats. PDF format now is truly cross-platform because it has full support from several vendors including free software projects. At the same time, *Word* and *LibreOffice* provide proprietary DOC format and free but not well known OpenDocument format respectively. These facts and especially uncompatibility of *Word* with Linux [24] make LaTeX practically the only true cross-platform solution for text documentation preparation.

3.2 Formulae Processing

Formulae typographic configuration is very hard task to be solved by ordinary office solutions which oriented on inline text editing model [17]. The WYSIWYG solutions are forced to divide the text editing and formulae processing into the independent tasks. In that case the formula is defined as *smart object* which has its own structure and processing rules. The special program producing such objects provides graphic representation of the formula as a bitmap image encapsulated (not always) with source code of produced object. This graphic representation can be interpreted as the special symbol within the text. In is not a vector image and requires to be re-rendered each time user changes its dimensions.

All known WYSIWYG office suites implements special technologies to create *links* from the text document to the formulae object code. Among the others, the office suites which use file format contains compressed binary data with hierarchic structure, like *Word* and *LibreOffice* provide most user-friendly interface. They allow to embed the formula directly into the document and to reserve required data space within the file. This technology, known as OLE[2] in Microsoft's variation [7,20,21] requires the special software called OLE servers for each type of object that can be embedded. Each one of them requires special integration into the specific solution. *Word* and *Microsoft Office* suite provides built-in OLE formulae processing server called *Microsoft Equation* (Fig. 2).

This model of formulae processing carries all disadvantages described above. Also it leads to impossibility to transfer the prepared document between even different installation of the same software because of possible difference between versions of the server (*LibreOffice* suite does not have this disadvantage because it provides encapsulated server named *LibreOffice Math*). The process of creation of a formula in visual environment which looks like shown on Fig. 2 is extremely slow because user is forced to permanently switch between keyboard

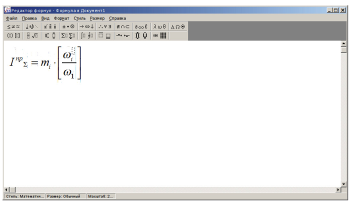

Fig. 2. *Microsoft Equation 3.0* interface

[2] Object Linking and Embedding.

input and graphically invokable commands that produce symbols and placeholders on screen. Practically the main and the only advantage of the visual formulae editing in the training process is good interaction between the object and user actions.

In LaTeX the formula is not an object [22]. It consists from usual symbols of the required fonts rendered into the geometrical structure [12]. The embedding of the formula does not required in this case. LaTeX renders it as plain text printed using special set of fonts. The formula rendered on Fig. 2 in LaTeX looks like:

$$I_{\Sigma}^{pr} = \left[\frac{\omega_i}{\omega_1}\right] I_i$$

and can be represented in document's source code as the following set of commands:

```
\begin{equation*}
I_\Sigma^{pr} = \left[\frac{\omega_i}{\omega_1}\right]I_i
\end{equation*}
```

The source text in the document contains formula divided into the set of mathematical objects presented as commands and expressions. The LaTeX core program performs rendering itself. Definitions inside the formula represent the consistency and sequence of calculations defined and ordered by statements of TeX language. This model of formulae processing allows to easily connect mathematical objects from text document to real calculation engine. For user the formula structure in LaTeX has very strong connection with structure of document itself. In training process it carries some very important advantages:

- Creation of the formulae in LaTeX is significantly faster than in WYSIWYG software suites because it is not necessary for user to switch between viewing of result and editing the expression.
- Sequence of expressions allows student to improve understanding of logic of calculations in TMM.
- Programmable structure of the formula allows student to easily use once prepared mathematical expression again and again, and even to build a database of formulae ready-to-use for further projects.

The only significant disadvantage for LaTeX formulae processing engine in TMM training is necessity to learn commands to build a formula. But existence of this disadvantage can easily be transformed into an advantage according to famous "Russian Method" [8,13] of engineering education. In this case the student improves his skills in programming, which also included in the course, simultaneously with performing the calculations by himself. After that he will have an additional skill which allows him to develop special project (IDE[3] or report generator for example) where the formula can be generated automatically using LaTeX commands, self-developed program and/or semantic database [15].

[3] Integrated Development Environment.

3.3 Graphics Creation and Processing

WYSIWYG solutions treat raster and vector graphics separately. Most common way to treat the raster images is to use embedding technology as it was described above. Rendering and unpacking of bitmaps are server's tasks in this case. For vector graphics there are special solutions integrated into the software suite [25]. These solutions perform limited operations with vectorized objects and all of them are embedding servers (Fig. 3) which produce raster graphics from vector object's source code before rendering it into the document. This method is absolutely incompatible with engineering education process because the obtained drawings can not be read using any standard CAD system or technical assistance software. Also WYSIWYG solutions contain no tools to satisfy the technical standards because it is not their primary function. The considered facts allows us to make a conclusion: preparation of technical documents containing graphics, especially vectorized, is very difficult using typical office software, even for engineers, not only for students.

Some solutions, like *T-Flex DOCs*, provide complex automation of technical texts creation and editing [10]. But for TMM training process these solutions are completely unsuitable because they all require special skills and authorized training courses for operators [16]. Within TMM course which completes in one year it is fully impossible to use such systems.

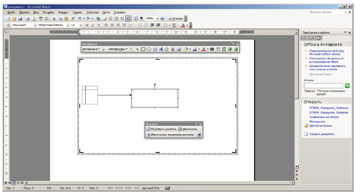

Fig. 3. *Microsoft Word 2003* OLE built-in vector drawing engine interface

In LaTeX the graphics rendering engine realized as set of modules which can be used separately. Each module, for example, `graphicx` [11] performs rendering of specific type of graphics. The possibility of direct rendering makes LaTeX engine truly cross-platform as it was already described above. But the most significant advantage of LaTeX graphic engine is its availability to render programmable vectorized graphics using scripts. Some modules like TikZpicture and PGFplots [18] allow user to create a script which defines plotting settings, including frame, legend, curve appearance. After that it is possible to transfer data from mathematical expression or file generated by another program (e.a. in CSV format), directly to the prepared script to render the plot. The short example shown on Fig. 4 renders 3D surface using mathematical expression defined as:

```
\begin{tikzpicture}
        \begin{axis}[view={110}{10},colormap/greenyellow,colorbar]
                \addplot3[surf] {-sin(x^2 + y^2)};
        \end{axis}
\end{tikzpicture}
```

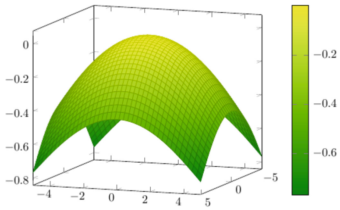

Fig. 4. PGFplots render example

The described possibilities make LaTeX practically the complete helper solution for preparation of technical texts including sophisticated tasks of generating vector graphics, script and array-organized data based plots rendering. During TMM training student can easily satisfy requirements of Russian method using experimental data and updating it directly while document preparation. It connects real development and training allowing university to improve alumni's competences.

4 Conclusions

Now LaTeX is in active usage in Bauman University as main free and open source text editing software. It replaces many proprietary solutions like *MS Word* and allows students to highly improve their understanding of engineering work using TeX programming, building of the calculation sequences and logical skills. The considered facts and peculiarities described above made LaTeX the ideal solution for text documentation preparation in Aerospace faculty of Bauman University.

Acknowledgements. Authors want to acknowledge prof. **Olga Egorova** for her assistance and advice during preparation of this paper.

References

1. Common development and distribution license (CDDL-1.0). Published by Open Source Initiative. https://opensource.org/licenses/CDDL-1.0
2. The MIT license. Published by Open Source Initiative. https://opensource.org/licenses/MIT
3. WYSIWYG, p. 1936. Springer, Boston (2001). https://doi.org/10.1007/1-4020-0613-6_21274
4. Erste Schritte, pp. 13–46. Springer, Heidelberg (2006). https://doi.org/10.1007/978-3-540-34584-8_2
5. GNU GPL general public license version 3, 29 June 2007. Published by Free Software Foundation (2007). http://www.gnu.org/licenses/gpl.html
6. GNU LGPL lesser general public license version 3, 29 June 2007. Published by Free Software Foundation (2007). http://www.gnu.org/copyleft/lesser.html
7. Bangsow, S.: Data exchange and interfaces, pp. 661–704. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-19503-2_14
8. Baryshnikova, O.O., Egorova, O.V., Omelchenko, V.: An efficient educational methodology for teaching theory of machines and mechanisms discipline. In: Proceedings of 14th IFToMM World Congress 2015, Taipei, Taiwan, vol. 1 (2015). https://doi.org/10.6567/IFToMM.14TH.WC.OS5.008, http://www.iftomm2015.tw/IFToMM2015CD/PDF/OS5-008.pdf

9. Belonozhko, P.P., Belous, V.V., Karpenko, A.P., Khramov, D.A.: Instruments for automated numeric estimation of trainees' meta competences. Sci. Educ. Electron. Sci. J. (10) (2015). (in Russian). http://technomag.bmstu.ru/doc/821623.html
10. Bocharnikova, N.A.: Complex automation of Uralkryomash AO using PLM-system. Autom. Ind. **9**, 7–10 (2016). (in Russian)
11. Carlisle, D.P.: The LaTeX 3 project. Packages in the 'graphics' bundle. http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf
12. Cohl, H.S., Schubotz, M., McClain, M.A., Saunders, B.V., Zou, C.Y., Mohammed, A.S., Danoff, A.A.: Growing the digital repository of mathematical formulae with generic LaTex sources, pp. 280–287. Springer, Cham (2015). (in Russian)https://doi.org/10.1007/978-3-319-20615-8_18
13. Egorova, O.V.: About "Russian Method" of training of engineers. Int. Aff. (1) (2014). (in Russian). http://interaffairs.ru/read.php?item=10459
14. Ghosh, R., Glott, R., Krieger, B., Robles, G.: Free/libre and open source software: survey and study. Technical report, International Institute of Infonomics, University of Maastricht (2002)
15. Groza, T., Handschuh, S., Möller, K., Decker, S.: SALT—semantically annotated LaTeX for scientific publications, pp. 518–532. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72667-8_37
16. Kochan, I.N.: Complex automation based on T-FLEX PLM platform. CAD Graph. **6**(224), 54–56 (2015). (in Russian)
17. Lin, Y., Zini, E.: An empirical study on implementing free/libre open source software (FLOSS) in schools, pp. 123–132. Springer, Boston (2006). https://doi.org/10.1007/978-0-387-37876-3_10
18. Norserium: Plotting graphs in LaTeX—PGFPlots (2015). (in Russian). https://habrahabr.ru/post/250997/
19. Roberts, J.A., Hann, I.H., Slaughter, S.A.: Understanding the motivations, participation, and performance of open source software developers: a longitudinal study of the Apache projects. Manag. Sci. **52**(7), 984–999 (2006). https://doi.org/10.1287/mnsc.1060.0554
20. Schön, D.: IT-Unterstützung, pp. 199–346. Springer Fachmedien Wiesbaden, Wiesbaden (2016). https://doi.org/10.1007/978-3-658-08009-9_5
21. Schwessinger, M., Schürmann, T., Süßer, K.: OLE und Dynamischer Datenaustausch, pp. 793–808. Vieweg+Teubner Verlag, Wiesbaden (1992). https://doi.org/10.1007/978-3-322-96369-7_42
22. Sofroniou, N., Hutcheson, G.D.: Technical typesetting using latex. Behav. Res. Methods Instrum. Comput. **26**(3), 369–371 (1994). https://doi.org/10.3758/BF03204648
23. Sorokin, M.O., Khoteev, S.D.: Free instruments for organization of information systems development in building. Youth Herald Tech. Sci. (1) (2013). (in Russian). http://sntbul.bmstu.ru/doc/532852.html
24. Vukolov, A.: Free and open source software applications for education of TMM discipline in Bauman university, pp. 253–260. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-44156-6_26
25. Wozny, M.J.: Beyond computer graphics and CAD/CAM, pp. 3–9. Springer, Tokyo (1987). https://doi.org/10.1007/978-4-431-68057-4_1