Příprava struktury programu na vyčištění typografických chyb pro Univerzitu Tomáše Bati ve Zlíně

Michal Polášek

Bakalářská práce
2007

Univerzita Tomáše Bati ve Zlíně
Fakulta managementu a ekonomiky

Univerzita Tomáše Bati ve Zlíně

Fakulta managementu a ekonomiky

Ústav informatiky a statistiky

akademický rok: 2006/2007

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Michal POLÁŠEK**
Studijní program:	**B 6208 Ekonomika a management**
Studijní obor:	**Management a ekonomika**
Téma práce:	**Příprava struktury programu na vyčištění typografických chyb pro Univerzitu Tomáše Bati ve Zlíně**

Zásady pro vypracování:

1. **Proveďte literární rešerši týkající se typografických norem a obecně uznávaných doporučení v typografii.**
2. **Zhodnoťte současný stav automatického vyčištění dokumentu od typografických, stylistických a jiných chyb vůči platným normám a vybraným doporučením.**
3. **Popište možná řešení pomocí softwarových nástrojů v různých DTP systémech, případně přes webové rozhraní u dokumentů webových.**
4. **Navrhněte strukturu a možnosti takového reálně zvažovaného programu pro eliminaci typografických chyb.**

Rozsah práce: **40 stran**

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Monografie

[1] GŰRTLER, Martin, KOCICH, Pavel. Visual Basic.NET. 1. vyd. Brno : CP Books, 2005. 312 s. ISBN 80-215-0367-6.

[2] HRADSKÝ, Josef, et al. Český Office 2000 Professional. 1. vyd. Praha : Grada Publishing, 1999. 476 s. ISBN 80-7169-882-2.

[3] MCCARTHY, Jim. Softwarové projekty. Praha : Computer Press, 1995. 190 s. ISBN 80-7226-164-0.

[4] MĚCHURA, Michal Boleslav. Vyšší škola Wordu. 1. vyd. Brno : Nakladatelství Šimon Ryšavý, 1996. 279 s. ISBN 80-902143-1-2.

[5] Microsoft Office Team. Microsoft Office 2003 : resource kit. 1. vyd. Brno : Computer Press, 2004. 628 s. ISBN 80-251-0177-0.

Vedoucí bakalářské práce: **Ing. Pavel Stříž, Ph.D.**
Ústav informatiky a statistiky

Datum zadání bakalářské práce: **16. března 2007**

Termín odevzdání bakalářské práce: **18. května 2007**

Ve Zlíně dne 16. března 2007

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta managementu a ekonomiky
L.S.

doc. PhDr. Vnislav Nováček, CSc.
děkan

doc. Ing. Rudolf Pomazal, CSc.
ředitel ústavu

ABSTRAKT

Tato práce se zabývá přípravou podkladů pro navržení a sestavení programu pro eliminaci typografických chyb, vyskytujících se v elektronických dokumentech zpracovávaných na Univerzitě Tomáše Bati ve Zlíně. V první řadě se jedná zejména o diplomové práce studentů.

Snaha o vznik takovéhoto typu programu není daná bezprostřední momentální potřebou, ale dlouhodobým opomíjením některých základních pravidel typografie, které jsou skrytým fenoménem každého psaného dokumentu. Tyto pravidla ovlivňují celkový výsledný dojem z dokumentu, jeho čitelnost, srozumitelnost, estetickou hodnotu a v neposlední řadě i jistý druh ergonomie. Typografie však není běžnou součástí školní výuky. A přitom dnes, víc než kdy jindy, proniká publikovaní rozličných typů dokumentů k široké mase běžných uživatelů. Rozvoj desktop publishingu (zkráceně jako DTP), neboli jakéhosi "publikování na stole", se mnohdy stal neoddělitelnou součástí nejen obchodní praxe, ale i mnoha jiných oblastí, z nichž vysokoškolské vzdělávání patří k těm zcela signifikantním.

Jistá a podstatná část informací určená ke sdělení obvykle skončí na papíře. Ať už je obsah těchto informací umělecký, komerční nebo jednoduše informativní, ve chvíli přenesení do psané podoby začínají podléhat pravidlům psaného projevu. Tato forma díky svému dlouhému trvání vyvinula množství pravidel, které slouží jedinému účelu – tj. úspěšnému a efektivnímu přenosu informace. Typografická pravidla jsou právě tím faktorem, který do tohoto účelu zasahuje masivní měrou.

Ve této bakalářské práci jsem se zaměřil právě na ona opomíjená, ale přesto velmi často používaná pravidla. Snažím se zde navázat na práci Ing. Jozefa Říhy, který před dvěma roky úspěšně obhájil a publikoval své diplomové téma: Tvorba elektronických šablon pro Univerzity Tomáše Bati ve Zlíně. Své úsilí zaměřil na velmi často se vyskytující druh dokumentů, které jsou tvořeny diplomovými pracemi studentů.

Výsledky jeho práce jsou v současnosti již v plném využití u všech fakult, pro než byly šablony vytvořeny. Bohužel díky velkému rozsahu projektu zůstaly některé jeho funkce a možnosti nedokončené. O této situaci jsem se dozvěděl díky upozornění Ing. Pavla Stříže (vedoucího diplomové práce Ing. Říhy). Ten mi nabídl spolupráci, kterou jsem přijal a jejímž výsledkem je tato bakalářská práce.

Obsah je rozčleněn dle zvyku na teoretickou a analytickou část. V teoretické části předkládám obecné základy typografie, vysvětluji okolnosti vývoje této disciplíny a konečně seznamuji s nejzákladnějšími státními normami, které mají přímý vztah k úpravě dokumentů a podstatě této práce. Dále seznamuji čtenáře s několika nejčasteji používanými programy pro tvorbu dokumentů nejen z řad populárních kancelářských balíků, ale i profesionálních softwarových řešení v podobě lámacích programů. Následuje přiblížení základů korektury, včetně vysvětlení principu softwarového řešení korektury překlepů, gramatiky a okrajově i stylistiky. Poslední kapitolou teoretické části je stručný úvod do rozdělení a teorie počítačových programovacích jazyků.

Při sestavování analytické části mé práce bylo třeba vycházet z obecných předpokladů řešení tématu, zcela konkrétních podmínek na výslednou funkčnost případné aplikace a zvláštních požadavků potencionálních uživatelů. Některé závěry, ke kterým jsem v rámci své práce dospěl, byly částečně předvídatelné. Jelikož by jejich ověření intenzivním dotazováním s následným rozborem výsledků bylo neefektivní, proto jsem pro formulaci svých závěrů použil takzvané „Delfské metody“. Charakter použité metody odpovídá spíše syntetickému šetření. Dotazoval jsem konkrétní úzký okruh programátorů a poněkud početnější skupinu náhodně vybraných studentů univerzity, tedy jak nositele technického know-how, tak i potencionální uživatele programu.

K formulaci mých závěrů také dopomohla neměnnost sledovaného prostředí, díky níž jsem mohl částečně vycházet z dat získaných Ing. Říhou (kapitola o výběru sázecího systému).

Obsahem analytické části je v první řadě stanovení prioritních požadavků na výsledný program z pohledu požadavků potencionálních uživatelů i funkčnosti. Pak již následuje pozouzení používaných systémů pro tvorbu dokumentů a srovnání jejich jednotlivých korekturních nástrojů. K této kapitole také patří srovnání s komerčním korektrurním programem. Výsledkem je výběr sázecího systému, s největším potenciálem uplatnění mezi uživateli a nejsnadnější aplikací softwarového řešení odstraňování některých typografických chyb. Následuje analogické šetření u výběru nejvhodnějšího programovacího jazyku. Stěžejní částí práce je návrh funkční části programu a jeho podoby, doplněný o doprovodný materiál vysvětlující některá základní typografická pravidla. Je přiložená i zevrubná kalkulace pro případ sestavení této aplikace na zakázku komerční cestou, která slouží pouze pro dokreslení finančního pozadí věci.

Téměř závěrečnou část práce tvoří návrh distribuce a několika dalších doprovodných služeb a informací, které svou povahou byly spíše doplňkové. Závěrečná kapitola proto obsahuje specifikaci uživatelského segmentu, podobu a obsah doprovodných informačních WWW stránek, způsob jakým lze využít interních zdrojů a konečně i způsob jakým lze do jisté míry garantovat zabezpečení aplikace. Ty nejdůležitější body práce jsou shrnuty v závěrečných doporučeních a návrzích. Tato práce nenabízí finální verzi aplikace, nabízí ovšem teoretický know-how a případnou snažší a rychlejší realizovatelnost konstrukce.

Klíčová slova: korekce, typografie, VBA, Microsoft Word, OpenOffice.org, software

ABSTRACT

The aim of this work is to prepare the starting documentation for the construction of typographical errors correctional software for Tomas Bata University. This work structure is given by the theoretical part, providing an overview of the typography basics, presently used typesettings systems, and their correctional equipment. The consequential analysis part evolves into the concrete solutions of typesetting software, programming language and application structure choice. There is also considered a suggestion of creating custom-made correctional software with only external sources in the end of this work. The paper work clearly indicates that this software construction under Tomas Bata University patronage offers a potential of successful realization and usage.

Keywords: correction, typography, VBA, Microsoft Word, OpenOffice.org, software

ACKNOWLEDGEMENTS

I would like to thank my mentor Ing. Pavel Striz for his helpful support and friendly guidance into the world of typography. Without his help this work would not come into the day light even in a Czech version.

Thanks to Petr Švehlík and those programmers of Sophics Ltd and Technodat Ltd too, for their precious time.

I would also like to thank my family and friends for their patience.

Thank you all!

CONTENTS

INTRODUCTION

This work is trying to continue where another one has stopped. Two years ago Ing. Jiri Riha successfully defended and published his diploma thesis. He wanted to create templates for electronic documents created and used at Tomas Bata University. His effort was pointed at the high frequently occurring document sort – thesis written by students.

The results of his work are in common use nowadays. Unfortunately according to the project size some functions and possibilities were left unfinished. I have started being interested in this fact by notice of Ing. Striz (mentor of Ing. Riha's diploma thesis). He gave me the first overview of this situation and offered me to cooperate. I willingly accepted. The result of this acceptation is this work.

This document is dealing with developing software for typographical correction of electronic documents. Its main purpose is much more about preparing the initial materials for software construction, than the very programming. The aim is to find out the best solution on answer how to do it possible and functional with minimal costs. I have decided for noncommercial developing with internal sources of Tomas Bata University. I am sure that there is some unused potential. There was established Faculty of Applied Informatics last year. I think that this kind of cooperation under our university patronage could bring some satisfactory results.

There is some sort of a gap in every document, even thesis creation. The gap is created by users, who are not aware of two things: the typographical basic rules and what is worse their national language basics too. Different text editor developers are trying to offer a treatment for this in general, but usually it is more a marketing play than complex and functional solution. The gap is still open. At the end of this chain there is one more weak link – a user not managing his software. So as a final result there are three affecting factors. The gap is open and waiting for someone to fall into. I just wanted to mark the sources of all possible mistakes with the gap comparison. It ends here. Present situation is not so emergent, but there is a strong potential of time wasting and inefficiency.

Some things are better to prevent. In the first, theory part, I am going to give some explanation of basic terms and rules accompanying this work objective. All statements found in a literature and other sources are giving the same or very close conclusions, so they are not offering almost any possibility of discussion or polemic. This is done by the sphere of interest, because the typography and typesetting are sets with steady rules proved by many years. I am going to use mostly deductive procedures to get my conclusions in this work. My intention is to collect all applicable rules of necessary spheres (i.e. typography, a Czech linguistics and Czech national standards) and offer the result in a user friendly pack. There are a huge number of all possible written mistakes that can be done by in a particular language. Application of one complex treatment in easy package is beyond this work possibility. Therefore I am going to separate only those rules, I will be sure of their practicability.

The analysis part is dealing with the best solution choice. There are going to be given the final software requirements, including the choice of typographic software, a programming language, a form of distribution, and a use. Although this software is meant to be noncommercial, the financial side is going to be checked for a potential expenses and profit.

I suppose this kind of typographical mistakes correction software is able to be created and successfully used for the benefit of Tomas Bata University. As I have found, there is a number of interested subjects in correct typography in Czech Republic, not only at our university. Unfortunately, those users, not aware of correct language and typographical standards, massively use some major office suites usually. Without any accessible tool or public enlightenment this reality will not change.

This work is only another first step towards the education.

I. THEORY

1 TYPOGRAPHY

A book, a journal, a newspaper… Even the smallest business card or giant advertisement has something in common. Every form of printed, painted, or some other way created document, stands on the basics of typography, although this is mostly not remarkable. If there is even a letter, a number, a sign, or some kind of layout, we can talk about the typographical influence. It is a wide application of basic rules.

Because the range of typography is very specialized and very creative, it is usually defined as an art or craft. The best way of understanding can be traced in history. The very beginning can be seen in the first punches and dies used to make seals and currency in ancient times. Later we are aware of developing the letter print in mid-15th century. Even though in 13th century there was some kind of letter printing with modular moveable metal type in Korea, the real expansion was created by Mr. Gutenberg's invention of moveable type printing. [1], [22]

It was a point of the whole typographic discipline to use written information effectively and aesthetically. The basic problem of printing was solved and had given a new artistic potential. Since this has been very demanding job, there is a wide specialization. Let us mention a typesetter, a compositor, a typographer, a graphic artist, an art director, and a clerical worker. That is the situation that has been generating last five centuries.

There were found some new ways and technological improvements. But another invention changed the familiar old way of typesetting very dramatically. The popularization and accessibility of personal computer in the eighties influenced almost every sphere of human activity. Not even the typography had left behind the progress. The basic rules remain the same, but the technological procedures changed. Thanks to the word processors the typesetting became mass available. A branch of typography called desk top publishing has been given brand new unimaginable possibilities. [17]

This is the age of the computers and communication. Any business is unimaginable without a good technology and information exchange. We do not realize that every time we use our computers for creating some sort of document for sending by email or just common printing, we are touching the rules of typography. We do not have to take care of many of them, but there are a few we should have on our minds. There is no reason to quarrel about it. No matter what, the sense is functional effectiveness of transmitted information.

1.1 Introduction to typography

To find out the explanation of the typography as an art, we have to look at the raw birth of the characters. It is a quite long way even from the thought to the alphabet. I am going to describe the old fashion method that used foundry, because it is very noticeable there.

The birth of a character

First of all there have to be a draft of the particular character. The drawing is transferred mechanically into the cubic template, which is used for creating a glyph later. These have been usually made in foundry of a lead. A cubic templates represented by glyphs are set into words and put in the lines aligned by the descender and page layout. This example is simplified in many ways, but it is going to be interesting now. [1]

Now the only one character is made, but we need much more characters for typesetting – a set of characters. To make it clear we have to begin from the basics.

1.2 Typographical basics

There is usually a single character in a form of an image. It is called a glyph. An amount of glyphs of one stylistic unity type is called font. As an equivalent meaning we can use an expression a character set. A set of fonts creates a typeface. If a body type is complemented with a style type and eventually other font of the same stylistic unity, it is called family of type. Diacritical marks often called as accents are a sign for chanting the pronunciation. A breve, an acute and others belong to this category. Diacritical marks are placed above, under or over the accented character. The accent makes with a character one complex graphical unit.

There are some very usual drawing changes used in the typography. The most noticeable changes are adjusted spacing between some characters and a ligature. The first one is used for adjusting the visual impress of black and white spaces. When those some characters are close to each other they produce darker visualization then other words on a page for example. This fact disturbs the reader and implicates aggravated readability. The ligature is joining character parts together, when they are in a contact. The result is graphically not disturbing and supports orientation in the line. We can observe that typography aims the reading comfort, efficiency and aesthetics. [17]

The discipline of typography has been developed through centuries. Therefore there are two main nonmetric measuring systems used these days. The first one is called Didot point system. It was improved by a French printer Francois–Ambrois Didot from older French Royal foot measure in the eighteen century. Didot point system is also titled as European. Anglo–Saxon measurement system is called Pica have been used by English and American printers. It uses an inch too, but these systems differ to each other. This difference is shown in (*Tab. 1*). The French pica is 4,512 mm although American printer's system uses a pica of 4,2175 mm therefore a compromise was set. The new computer pica is defined as 4,2333 mm. This becomes a new standard in DTP. [17]

Tab. 1 European and Anglo–Saxon measure system comparison.

Didot point system	***Pica system***
French Royal foot = 32,48 cm	English foot = 30,47 cm
1 point = 0,3759 mm	1 point = 0,3528 mm
12 points = 1 cicero = 4,513 mm	12 points = 1 pica = 4,23 mm
2660 points = 1 m	72 points = 6 picas = 1 inch = 2,54 cm

Because the measurement system uses pica and point there is a distinction between them in script. The picas are represented by uppercase „p“ or „pc“ character and points signify lowercase „p“ or „pt“ character. For example „8P10p“ or „8pc10pt“means eight picas and ten points. In Czech terminology is used „pt“ for pica and „b“ for a didot point system.

The result of this typographic evolution is using of three measuring systems together. Metric system is usually used when we talk about page, columns, and image size. But the font size is defined by typographic points. [1]

The font size is defined by a size of a glyph. The size can be determined by the scheme of lines. Each of them has specific name and purpose. It does not matter if a character belongs into lowercase, uppercase, superscript or subscript category. Even numbers, diacritical marks, punctuation symbols have their specific rules on positioning on these lines.

The lines are determined as:

- baseline – the characters are placed on in a row,
- meanline – determines the high of lowercase characters,
- ascender line, capline – determines the highest point of lowercase and some uppercase characters for some fonts, the uppercase characters are meant without diacritical marks. For example „b, d, f, h, l, t, …“ or „T, D, A, …“,
- descender line – represents the bottom end of the lower character‘s part.

The space between baseline and meanline is called x-height and affects readability and lightness of the line, if you like the page lightness. This aspect is shown by (*Fig. 1*).

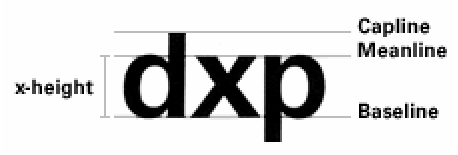

Fig. 1. Character's x-height[1].

The character structure is given by different parts. These parts are common for every type family and it is an important part of typographical nomenclature (*Fig. 2*).

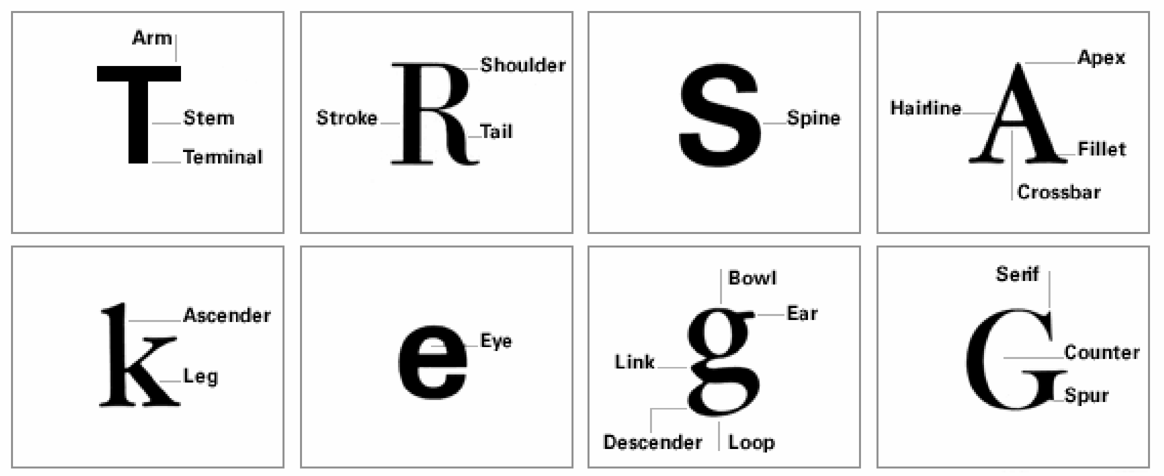

Fig. 2. Characters structure with different font type[1]

Normally used upright form of font is called roman or regular. For emphasizing a type styles are used. Recommended emphasizing for texts in roman is italic (also called cursive, italic types or oblique), weight of faces (semi bold, bold, heavy, thin, light), bold italic (a combination of slope and weight) and caps or small caps. This is quite important information for regular user, because a modification of underlining is still heavily used, although is beaks the graphical look of a page a lot. [1]

[1] The parts of letterforms [online] 2007, [cit. 2007-04-08].
<http://glass.typepad.com/journal/2003/12/the_parts_of_le.html>

Typographical standards

Historically there are many typographical standards used today. But some of them affect this work essentially. Typographical standards correspond very closely to the national language. Therefore there are some specific differences for every language.

There had to be done agreements because of successful international cooperation. National standards were discussed and fully new applicable standards were developed for worldwide use. The exemplary representative is International Organization for Standardization (ISO) founded in 1947. Its documents are copyrighted by law. Some of them are charged others are free. This organization helps to describe products and their parameters. Czech language is considered to be the one of more difficult. These standards were made by Czech standards institute for publishing text in Czech environment:

- ČSN 01 6910 Úprava písemností zpracovaných textovými editory (Guidelines for text presentation), released on IV. 2007,
- ČSN ISO 1000 Jednotky SI a doporučení pro užívání jejich násobků a pro užívání některých dalších jednotek (SI units and recommendations for the use of their multiples and of certain other units), released VI. 1997,

These standards shown above are adjusting only graphical layout of some document parts (i.e. punctuation marks, abbreviations, marks, numbers, emphasizing and structuring of text, columns and tables). But the amount or rules is influenced strongly by Czech spelling rules adjusting the use of language and its particular sphere.

1.3 Typesetting software

When we talk about typesetting software we have to be aware of the difference between the typesetting software and word processing software – text editors. Desktop publishing (DTP) is a discipline coming out of the typesetting and possibilities of computer processing. It concerns rather all document preparations but printing. From the point of simplicity the text editor is easier to handle. It offers fewer possibilities used for creating a currently occurring modular paper documents (reports, letters, manuscripts …). Options are customized for quick access, modification and production in low quantity. Those documents determinated for the larger number of prints are finally transformed into professional typesetting software. The conclusion is that the word processor belongs into a typesetting software category, but it is on much more lower technical level. The result is readable and acceptable. [17]

On the other hand the professional typesetting software should offer the most possibilities of the document modification at the top high technological quality. There were mentioned the typography as the art a few pages before. This is the same case, because between all these options only an art or a feeling for aesthetics can excel.

For the purpose of this work there are two main categories as follows:

- commercial and noncommercial,
- professional and common use.

This is going to be useful later in the analysis part. There is no way of telling which software is generally the best. All the products were made for a different kind of a final customer with diverse needs. But it is possible to choose the best for the ahead known aim. There is any software containing only the basic options because nobody would buy it (supposing it is commercial software). Even a noncommercial product has its standard of options so there is no basic software category.

Next few pages are going to introduce the most used typesetting systems at Tomas Bata University. This information was put together from [23] and personal interviews with students of different faculties.

1.3.1 Microsoft Office

Microsoft Office is a suit of productivity programs created or purchased by Microsoft Corporation. It has been developing for well known operating systems: Microsoft Windows, Apple's Mac OS and Mac OS X. since nineties. It was oriented mostly the business users at first, until the personal computer became worldwide available for a large number of consumer sectors.

A version for Windows operating system made its appearance in 1990. The main selling point was that buying the bundle was substantially cheaper than buying each of the individual applications. The first version of Office contained Microsoft Word, Microsoft Excel, and Microsoft PowerPoint. Over the years the Office applications have grown substantially closer together from a technical standpoint, sharing features such as a common spell checker, OLE data integration, and the Microsoft Visual Basic for Applications scripting language. [27]

The Microsoft Office suite is still very popular software standard these days. It is done mainly by Microsoft Word and Microsoft Excel. These two programs are joining together a word processor and a spreadsheet. The result is the base for quick publishing at home or office. Microsoft Word possesses a dominant market share in the word processor market. It offers a nice graphical user interface (GUI), WYSIWYG system, familiar look of Microsoft Windows operating system and language localization.

The predestination is not professional publishing use. The typesetting of more sophisticated or large documents is possible, but the typesetting options are limited.

Microsoft Word is a commercial product. Its price differs from a product to product. It is only available in MS Office suit. These come in large variations. The price for the last MS Office 2007 system is 3,000 Kč – 14,000 Kč depending on a license type.

1.3.2 OpenOffice.org

OpenOffice.org was created in 1999 as a noncommercial reaction for the dominant market share of Microsoft Office. From the beginning there was the point of providing free alternative to the other major office suites. OpenOffice.org is a product trademark of multiplatform and multilingual office suite and an open-source project. It can work on Linux, Microsoft Windows, Solaris, OpenVMS, IRIX and Mac OS X based platforms. Compatible with all other major office suites, the product is free to download, use, and distribute.

The source code is written in C++ and delivers language-neutral and scriptable functionality, including Java(TM) APIs. This source technology introduces the next-stage architecture, allowing use of the suite as separate applications or as embedded components in other applications. The software license is the GNU Lesser General Public License. OpenOffice.org binaries are legal for commercial use. [28]

Propagation is aimed on fair relationship with customer. All the source code is open.

The latest OpenOffice.org 2.2 version contains:

- Writer – a word processor
- Calc – a spreadsheet
- Impress – a presentation creating software
- Draw – a graphic software

- Base – a fully functional database software
- Math – an equation editor

GUI is very friendly although it does not come from exact classical Microsoft schemes. Its working system is WYSIWYG. The look is pleasing and not confusing. The typesetting options are almost the same with the competitive products of Microsoft work. The suite has been translated and configured to at least 45 languages besides English.

OpenOffice.org gains more happy customers lately because of approach to a customer and zero purchase expenses. It offers a new way of cost minimization. There was added a graphical support with version 2.0 for different platforms to better match native applications and provide a smoother experience for the user. [28]

1.3.3 TeX and LaTeX

It is a professional typesetting system created by professor Donald E. Knuth from Stanford University in early eighties. Its main feature is precision dealing with every kind of mathematical content and text. This software is public domain which means it is distributed fully for free and the source code is open to everyone. Therefore there is a pretty widespread base of its fans and volunteers developing this software on continually. The only thing mattering mass usage is a starting demand for user‘s knowledge.

The first version was named TeX (The name comes from Greek. Its primal means a technology and an art.). It was built so well that it has not changed essentially up to the present day. The core is very opened to the expanding of its facilities.

For the purpose of professional typographic result many functional overlays were added to the basic TeX software. A very popular overlay is called LaTeX. It is a document markup language preparing source document for the processing by TeX. An input is made of the plain text and information about attached graphics files. These are filled in with document and text formatting commands. Syntax is very similar to the HTML language. The processing output is a *.dvi file with its content in type. It has to be processed with another transformation program to be able to be printed or seen on screen. [17]

The range of application expands thanks to the modifications called packages. They have to be joined to the processing by a command is document structure. The package is a functional addition of new possibilities or improved features. They are dealing for example with colors, graphics, different languages support, fonts, document structure, hypertext, output formats and many others. Each package is usually specialized for a particular feature and has its own documentation.

Work flow in TeX is different because it is non-WYSIWYG; it is quite complicated for basic users used to graphical interface only. Although there are some graphical add-ons, this fact is for the most users very hard superable. Advanced and professional users are highlighting program‘s plentiful typesetting options, so this thing is not apprehended as a disadvantage by them. Popularity is noticeable amongst the academical users, especially in the mathematics, physics, computer science, economics, political science, and engineering communities [11].

Typesetting system TeX is efficient for advanced users demanding high quality tool. A reward for the invested effort of mastering this system is quality of program outputs and zero purchase expense.

1.3.4 Adobe InDesign

Adobe InDesign differs from the previous office examples. It cannot be compared with the office suits, because it is first of all dedicated to a professional publication of high quality documents. This is a truly pure DTP commercial system.

It can run on Microsoft Windows and Mac OS X based platforms. If we talk about TeX as an ideal solution for text and mathematical content, Adobe InDesign is the same one for text and graphics. The WYSIWYG work method and graphics possibilities of created document are not the main benefit. It is technical accuracy with the accent on rich creative options. This DTP system is not only used for book publishing, but also for newspapers, magazines, and other publishing environments.

Adobe Systems company produces large number of software. But there is a focus on dealing with graphics. Adobe made the account of this fact; thereby InDesign closely cooperates with Adobe Photoshop and Adobe Illustrator. Finally it makes a big set of graphical tools with similar interface and technologies.

There are available many modules extending the possibilities of Adobe InDesign. Those are usually products of third-party developers. It covers more or new possibilies of database cooperation, lists, cross references, new graphic features and many others.

Adobe InDesign is a commercial product for professional DTP use. GUI is well known from other Adobe products. The price for the last InDesign CS3 is 22,800 Kč – 30,000 Kč.

1.3.5 QuarkXPress

QuarkXPress is real professional typesetting software. The first version was released in 1987. The Quark, Inc. is considered as a founder of Desktop Publishing (beside Aldus, Adobe Systems and Apple Computer). There are available versions for Mac OS X and Windows. QuarkXPress was the first which shown the computer typographic way and it was the main reason for a direct competitor to develop the Adobe InDesign software.

QuarkXPress early incorporated an innovative application programming interface called XTensions which allows third-party developers to create custom add-on features to the desktop application. It was the same way, Adobe and other software developing companies are putting to use these days. QuarkXPress comes in different variation too. It is called QuarkXPress Passport containing of the same software as mentioned in the name, but with the added ability to use multiple language UI, hyphenation and spell-checking. [29]

This software offers all possibilities of professional typesetting product. New featured add-ons can be purchased from third-party developers. The latest version QuarkXPress 7 is selling for price 12,000 Kč – 30,000 Kč depending on software version.

2 SPELL CHECK

A language is a complex system of communication. There are two forms of language, depending on its character. The first one is spoken. It consists of different sounds and gestures. This concept can be transformed into a written form, the second one.

The public judges the knowledge of spell checking as a significant display of education. There are expectations in consistent and general complying with check spelling rules for preservation of a nation and its language. Therefore creation of such quality and functional rules is expected. It should be straight not admitting another possibilities. [6]

This written form experienced real boom in computer processing. At the same time a new request was found. There was a massive demand for automatic spell checking. This new demanded feature was perceived by software producers as a tool with strong marketing potential. Although the improvement of this is always advertised, there are still some limits of spell checking. The known example is shown by appendix P I.

This example is dealing with a poem in English. The words are spelled correctly, but the text is put together only by sound of its pronunciation. This kind of text is absolutely invincible handicap for simpler common spell checkers.

Spell checking in Czech language is not the easy one. Czech language expressing is based on details with words and character‘s variations. The sentence construction can be very voluminous and complicated even for the native language speaker. This covers a great potential of language use (e. g. belles-lettres), but also escalates spell checking software requirements. Anyhow the result is that spell checking feature is usual standard function of software processing some text data for publishing (not only DTP or text editors). More detailed information is going to be given in analysis part.

In principle, all spell checkers are working the very similar way. It compares its internal word database with the words in checked document; this is usually done with morphological description. After a text is analyzed and divided into single words, it is tested for similarity of characters and the most complying substitutes are offered. Some changes can be done automatically, even during data entering. Advanced correction software allows performing a grammatical analysis of a sentence. These corrections can be done only by given clauses and programmed algorithms. These products are mostly commercial. [7], [24]

2.1 Typing and random errors

This kind of errors is created during data entry and data formatting. There is only one factor appearing with data formatting. It is user's knowledge of managing his software functions. The lack of basic typographic rules knowledge is getting a quite difficult situation with all possible software options. User is equipped with many tools unfortunately he is not aware of the correct utilization. The only gauge for his action is the graphical result, which depends mostly on user's feel. It can be sometimes very far from readability and graphical balance.

On the other side entering data is a process based on mechanical and psychical interaction [18]. Therefore there appear more factors with data entering. In general they can be distinguished into two main categories:

- attention (eye contact, correcting, speed …),
- movement (typing on keyboard or moving mouse, accuracy …).

The setting of software offers many automatic corrections, but it has to be set correctly. Wrong settings can even take effect in false text modifications. This fact makes using text editors counterproductive and leads into the depressed effectiveness.

2.1.1 Grammar and stylistics

There are some limitations of text correcting. Even thought software can be programmed for complex dealing with spell checking and different kinds of text corrections, it can not involve every lay of language use. Contemporary text correcting software exploits internal word databases, thesaurus and consecutive analysis. It is not hard to decide about word spelling of the one particular word and offer some suitable solution to the software user, but it is very complicated to analyze a sentence or structure with all possible meanings. Sometimes even a sentence has do be corrected in conjunction with other text content.

2.1.2 Difficulties of automatic check spelling

Automatic can be defined as operating with little or no direct human control, but automatization of text correcting is a little bit complicated problem. A text analysis procedure is not capable of independent activity; therefore an interaction with software user is needed. As a result text correcting can not be fully automated, because the number of user's interventions is getting bigger with the text volume, complexity and finally a number of all misspellings done in text. [18]

2.2 Proof-sheet use

Written language use is subject to rules. The most general are typographical rules, covering formal text construction. The form of words is covered by spell checking. The discipline of revealing, pointing up and removing mistakes is called a proof-sheet. It deals with all possible kinds that can occur in document content.

In the 19th century the trend of standardization affected Czech proof-sheet too. There were established ČSN 88 0410 Korekturní znaménka pro sazbu – Pravidla používání (Proof correction symbols for printing and their use), last update released on VII. 2004 and ČSN 88 0411 Korekturní znaménka pro reprodukci (Proof correction symbols for reproduction), last update released on VII. 2004. These standards became the main set of rules for the branch of competitive text transcription called text correction.

This discipline has been developed from stenography. It has had a form of typewriter typing at first and computer writing later and today. Some branches are specialized for text managing with text editors including proof-sheet instructions. The periodic competitions are organized by international organization Intersteno (est. 1887) every two years. The Czech republic has a very prestigious status in these competitions because of contestant's quality. The number of medals for the Czech Republic was always nearly fifty percent during the last two world championships in 2003 and 2005. This year the world championship takes place in Prague on July 2007. Last time it was held by Czech organization before 44 years.

Although there are the competitions of text correcting, it is not commercially successful for advertising the need of right Czech spelling and need of correctly written language. The common user of Czech language is dependent mainly on school education.

The document correction and proof-sheet has become a business. When the correct use of Czech language is needed, there are a lot of individuals and companies specialized in different language document corrections. The correctional charge is usually defined by document language and the amount of corrected text, expressed by standard sized page.

3 PROGRAMMING LANGUAGES

Every personal computer is a collection of dead pieces without software equipment. There would be no action or processing without compiled language, that hardware base can fully understand. Therefore the programming languages were created. They are defined as the special and very specific languages, which computer understands at the certain level. [16]

The simplest way of computer language is binary code containing of ones and zeroes (So called binary digit, therefore it has abbreviation 1 bit reflecting its size.). It is equal to do and do not. It is the very elemental and primitive possibility of information transfer. This is the base the other languages build on. Although it is very easy, it has huge quantity needs (Let us just remain that old 3,5‘‘ floppy disk with capacity 1,44 MB has 2 097 152 bits.). Although the binary code is the only one that processor operates, there must have been created something better to program with.

Tab. 2 Basic classification of programming languages

Programming languages		
Low level	***High-level***	
	sequential	***object-oriented***
binary code	Basic	C++
assemblers	C	Visual C
Z 80	Fortran	Visual Basic
MASM	Pascal	Visual Java
…	…	…

Many different languages have developed during the time (From the low-level languages working at the hardware or operation software level, to the high-level ones dedicated for creating self-standing applications.). The basic classification can be seen in (Tab. 2).

The low-level language is characteristic for more or less cooperation with operation system and hardware. It is easy for the programmer to open drive or check bite size at the printer‘s port, even a manipulate with disc sectors. These operations could be very difficult with high-level languages, actually there are impossible sometimes. Another advantage is shorter processing time.

There are not only advantages. These low-level languages are hard to understand and their syntax is far away from spoken languages, so they are even hard to learn, to remember and to use. On the contrary high-level languages are more user friendly. The price is certain – longer processing time and fewer options. [16]

High-level languages are divided into two categories by their processing approach. Sequential programming is characteristic by its logical structure. It is written from the beginning to the end. The instructions are performed one by one except for loops and conditioned skips. The routine is done after the last instruction is processed.

Object-oriented programming does not have to be written in the exact order, because it contains of short procedures. Those are started by the defined event. After the procedure is done, the program is waiting for the same procedure to be run again. This kind of program is in fact endless, that is why its termination has to be provided by the program author. [14]

Use of programming languages

Every programming language was created for a specific use. In general it is possible to say that there is a kind of predetermination. It is done by the language's character and potential. Possible applications of programming languages:

- self-standing,
- a part of some current application,
- server-side,
- client-side.

The last two possibilities are mainly exploited by WWW, but it can also operate on a single computer with necessary server software. Fortunately there are many software products free of charge for public use. I have my personal experience with EasyPHP[2] suite which covers an Apache server, a fully PHP execution and a MySQL database.

[2] Actual information about EasyPHP software package:

http://www.easyphp.org/

II. ANALYSIS

4 FORMULATING SOFTWARE REQUIREMENTS

In order to create software there are basic steps that have to be done before the programming. There are mentioned the most occurring problems with software projects in [15]. From nine critical problem causes two of them are crucial. These two are:

- insufficient analysis,
- wrong assignment.

One leads to another. With wrong information from opening analysis there comes a misconception. Assignment comes from the user requirements but also programmer's limitation. This work is preparing materials needed for software assemblage; therefore the priority will be user's requirements. Programming language is going to be examined with the view of usability and its limitations in individual chapter.

Software requirements

I am going to list some ideas of mine. These ideas are a result of discussions with mentor Ing. Striz and my student colleagues. I have asked them for their opinion on typographical errors and practicability of such a kind of correction software. Those discussions with my colleagues were not standardized and were in progress casually. Therefore I have no exact information for statistical interpretation. My questions were usually open and they were formulated during interview. The number of asked individuals is near to 25. They were mostly represented by students of 2nd and 3rd year. Only few responders were from another faculty than Faculty of Management and Economics.

As I have found all answers had the same analogical conventional attributes, thereupon I came to an inductive conclusion. The questioning of larger number of responders and subsequent data evaluation would lead to the same result and would be anticoincident to the made effort.

The analogical attributes were: *simplicity,*
availability,
low time-consuming manipulation.

With a base of these three requirements I have defined three starting requirements and after that I have added another five ones. Finally there are eight requirements for the software.

They were formulated as follows (sorted by their importance for a user):

- functionality – The software should be able of correcting frequently occurring basic typographical mistakes in Czech written documents. Its sphere of action is given by the mistakes not covered or covered only partially by another correction software. The software should not be assembled for the number of possible covered mistakes at the expense of stability.
- availability - Although the use is very specialized, it should be conformed to the biggest amount of users. This software should be available for free download on the internet. The GUI should be user-friendly, systematic and brief as possible, but with all possible options and settings. On the other hand program documentation should be complete, bilingual (in Czech and English for the Czech language corrections). The documentation should be illuminating, including text and graphics.
- noncommercial product – A noncommercial form of distribution has the biggest potential of a massive use. It is easier to obtain larger support from public and expert community, because there are minor expectations of software under development. No price (or low price) increases user's interest and decreases entry barriers for him to fully use the software.
- security – This software should guarantee its security for not-allowed modifications of a third-party. All source code modifications should be controlled. This can be done by official central location for download and CRC checksums verification. Every manipulation with user's data should be performed on document copy this has to be provided in first software process steps. A user should be warned of modifications made in the document copy, eventually for the mistake location, type and quantity.
- enlightenment – The purpose is not only to come out with software solution, but also instruct the user of some basic typographic rules of his national language. This software is first and foremost aimed to the Czech language corrections. It can be modified for another national language, but this has to be followed by adjusting to another language standards.

These last two requirements are given by the financial side of this software development, although they are nearly invisible for user's interest.

- low expense of development – The financial factor is not important from the view of maximizing profit, but from the view of minimization the expenses. This software development is not a complicated project. Different parts can be done very quickly in two or three phases into a functional prototype. This development is based on voluntariness and non-profit cooperation.

- use of internal sources within Tomas Bata University – This is not only an economical factor. The aim is to offer a meaningful assignment as a support of classwork or self-education. This is able of joining a potential knowledge of students and lectors from different faculties, especially Faculty of Applied Informatics. Apart from the fact that Tomas Bata University could offer an eventual covering of software patent protection (providing that the software is assembled and functional).

5 CZECH SPELL CHECKING SOFTWARE

The purpose of this chapter is to chart the mostly used correction software and its possibilities at Tomas Bata University. This can be easily done by exploring the most used typesetting systems, because every kind of this software is equipped with some sort of language and typographic correction options.

5.1 Spell checking software included in other software

These options are usually different from product to product and there is no standard regulation for their comprehensiveness. Therefore their possibilities are quite different. Our students are aware of this fact. As I got to know some students from other faculties, especially Faculty of Multimedia Communications, are used to correct their desired text with different software, even if they are going to create their document with another software. For example about InDesign CS2, it is a part of their curriculum, I know that it does not allow unbreakable spaces (it can not end the line, it joins two words together) from participating in word spacing for Czech and Slovak languages.

These little limitations lead into using the most known correction software from Microsoft production sometimes. Although it is a part of office suite with Czech spell checking not all correction are taken into account. Here comes a comparing review of correctional software included in the three most used typesetting systems, as shown later.

5.1.1 Spell checking in MS Word

Microsoft is a company with dominant market share at office software products. Its products become an unwritten standard of provided services. Almost every two or three years there is a new release of MS Office an office suit. Microsoft tries to increase the product quality and its functionality. In the case of spell checking and another document correction in Czech language, we can talk about not so outstanding progress.

From the first MS Word version there were included the very basic spell checking and automatic corrections. Every new release was enhanced with a few more options. The only radical change can be seen in June 2005 when the Czech grammar checker for MS Office 2003 was released. It is true this software is focused mostly on the grammar and sentence construction. Let us take a look at the functions offered in a previous few years.

Standard correction options in MS Word

These options were present at the version MS Word 2002. I have chosen this version as a predecessor of those released later with improved grammar check. I have checked the options on my home version of MS Word and versions installed in TBU library too. All options were installed fully and functional in both locations.

Spell checking is limited only to an easy comparison with an internal word database. Automatic corrections are as follows:

- replacing "English" quotation marks with the „Czech“ ones,
- replacing English ordinal number with index,
- replace the space behind prepositions "k", "s", "v" and "z" with unbreakable space,
- replace two hyphens "--" with em-dash "–" or if the hyphen is surrounded by spaces,
- replace first two uppercase characters in one word with the only first one,
- create an uppercase character at a sentence beginning,
- replace specific written text with another from a custom word database,
- …

Others options are mainly focused on style formatting. The only optional typographic correction settings can be seen in handling with the unbreakable spaces behind some prepositions and difference between a hyphen or dash recognition. The MS Word 2003 offers almost the same correction options. The only upgrade I have noticed is a functional hyphenization and improved thesaurus.

Czech grammar checker for MS Office 2003 suit

Microsoft evidently has known of this missed feature and as a reaction started developing correction software with cooperation of Czech language institution. After five years of participation Microsoft came out with Czech grammar checker for MS Office 2003. This application was presented additionally to the office suit and from now on it is going to become a standard supplement for Microsoft's further Czech localizations of MS Office.

New added features were focused on functional gramatics checking and some new typographical corrections were implemented as well. A checked document was used to single out recognized errors by underlining them with a red wavy line. This was used for spelling errors. There was a requirement for highlighting grammatics errors too, so a new use of graphic markings were introduced. A green wavy underlining symbolizes a grammatics or stylistics error and a red underlining is still with the same meaning of misspelling, in other words a presence of the unknown word.

The new typographic or related corrections are:

- missing or incorrectly placed comma,
- highlighting of some misspelled abbreviations,
- accumulation of the spaces or a missing space,
- accumulation of punctuation marks,
- redundant space with used punctuation marks.

There was mentioned at the press conference on the occasion of Czech grammar checker release, that this tool is not a substitution for school education, but it is more likely a highlighting of some omissions. This approach to the correction software has only one big deficiency when the omissions are some not covered errors. Unfortunately those errors are still present at the document and a user is not aware of them.

5.1.2 Spell checking in OpenOffice.org

A development of OpenOffice.org was easier from the point of demanded software functions. Those were designed to be up to the competitive product functions of Microsoft. The OpenOffice's interface is very similar to the MS Word one. All functions are pretty the same and located as usual so a user do not have to grope for the function he is used to. A navigation bar is equipped with different icons and some menu is a little bid different, but still easy and manageable by instinct.

I have tested the latest version OpenOffice.org 2.2 for the correction options. I can say there are only those functions I have mentioned about MS Word 2002. An effort to equilibrate the standard functions was probably the main aim instead of overtake the competitor's offer. After all there are some new functions, but their importance is marginal.

Vlnka 0.4 – corrector of unbreakable spaces for OpenOffice.org

In comparison with MS Office, OpenOffice.org has a very good support of public user’s attitude. Thanks to this fact there are many add-ons created by different users and the software is getting sophisticated. These add-ons are designed for public use and they are increasing the software utility. One very good example can be seen in the Vlnka, an add-on application developed continuously by Robert Vojta. This application replaces particular spaces in text with unbreakable spaces according to the Czech typographic customs. These options are available (place unbreakable space in the following cases):

- one character preposition or conjunction,
- two characters preposition,
- units (kg, cm, …),
- a group of numbers,
- a numerical written date,
- a degree or an salutation,
- a centigrade degree or perceptual expression.

Except these options, an elimination of space at the end of a paragraph and blank lines deletion at the end of a document are offered. This makes the most powerful tool for correcting typographical errors in Czech written document. I have tested Vlnka 0.4 with the latest OpenOffice.org 2.2 Czech language version. Installation was smooth and quick. After twenty seconds the application was ready to use and fully functional. Unfortunately this solution is only functional with the OpenOffice.org software.

5.1.3 Spell checking in TeX and LaTeX

The multiplatform use of TeX is a positive fact for advanced users; unfortunately it is not so good for new ones. The advanced user is celebrating potential possibilities a beginning user is getting lost in them. The TeX was designed for users with passion for a professional result and the software substance is far away from an easy instinct operating. Spell checking is not only managed by one application. There are many spell checking applications for TeX and they are modified for different operational systems. There are usually more than two applications per an operational platform. Some spell checkers are made for a use MS Windows environment and they are incorporated into the shell/editor combinations that are available. There can be used a separate package 4spell or an editors TeXnicCenter or Winedt. As I have found only 4spell supports easy Czech spell checking. Its Czech word database contains nearly 170 000 words, nevertheless it is only basic help and other possible mistakes (grammatics, stylistics and typographic) are not covered.

I can say that TeX and its following variants have the same supportive users as well as OO.org. The TeX users were the first who came with modifications for automatic inserting unbreakable space by the Czech language use and standards; those were called Vlna and Vlnka. It was for a diacritical mark (~) used in TeX for the unbreakable space. That first version was dealing only with one character prepositions. The situation has changed only decently.

An amount of satisfied TeX users is limited by the GUI and non-WYSIWYG processing. TeX is not offering intuitive and easy interface for common use, and there is only a small number of users willing to learn to start using TeX successfully.

5.2 Commercial spell checking software

There is only one official Czech language correcting software these days. It is a commercial product Grammaticon. Although it is the most powerful tool for language correction, unfortunately it does not cover any typographic mistakes. Apparently the knowledge is expected from user and software function is dedicated only to the language side.

The Grammaticon offers spell checking, also grammar and stylistics check. By the review in [24], the possibilities are slightly better than Czech grammar checker for MS Office 2003 (mentioned a few chapters before). It can be used as a functional easy text editor or an add-on to every version of MS Word. It is primary designed for MS Windows, but it can be easily recreated for another application or operational system. These are only a few of its plentiful facilities.

From the point of function this software offers top services about Czech spell checking for about 1,000 Kč. Its commercial character is fully balanced with the software potential, but it is markable that one thing seems to be missing. The typographical corrections are limited only to punctuation symbols, other errors are still left uncovered by this product.

5.3 Typesetting software – the choice

From the information given above OO.org seems to be the best equipped with typographic correction possibilities. Unfortunately, in this case, the best equipped means equipped with a few more options above the other‘s products average, thanks to the effort of an individual. As I have got to know Ing. Riha did a survey of the most used typesetting software. His survey [23] took place two years ago and was concluded with the expectable final balance.

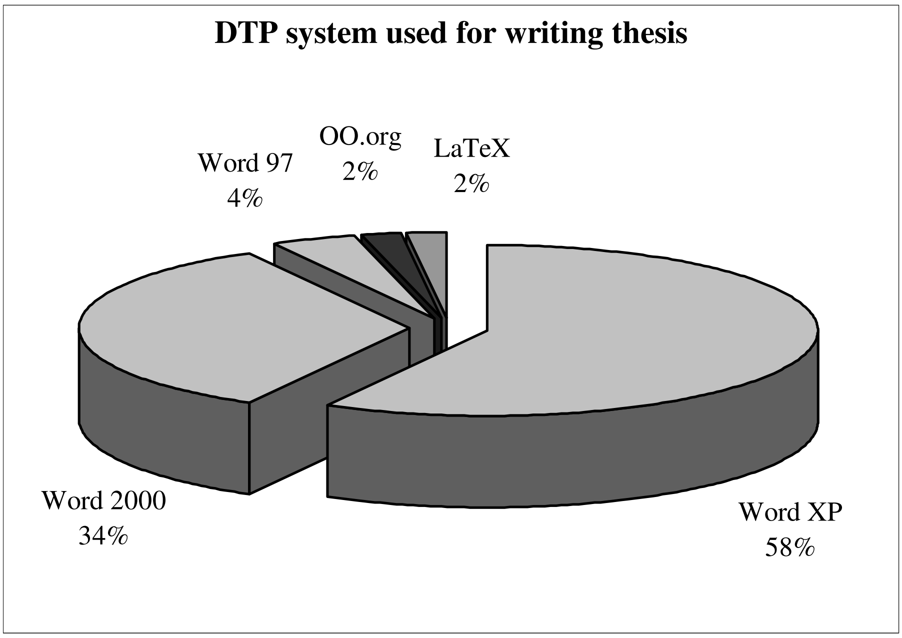

Fig. 3. DTP system used for writing thesis [23]

Although these figures are two years old, thanks to the stable market situation, we can expect very similar result. The most changes are connected with a migration to the newer MS Word versions. The ratio of used systems is not going to change dramatically. All computers in university library are equipped with MS Word and even the classwork of computer science at faculty of management and economics is centered on Microsoft products. The propagation of noncommercial alternative software solution is not emphasized enough even it can become an advantage in a competitive environment.

For the purpose of creating the application for correcting typographical errors, it has to cooperate with the latest MS Word versions. The use is supposing MS Windows platform. That kind of implementation warrants the biggest possible range of potential use for the Tomas Bata University or public users.

6 PROGRAMMING LANGUAGE

This chapter affects the way the program is going to be working. In the case of the typesetting system there was a requirement for as massive use as possible. Here is the aim quite different, the choice of programming language is about an easy applicability. The processing speed, time needed for an assemblage, and final form of software are going to be influenced by this step.

6.1 Programming languages – a description

There was chosen a typesetting system for the best user accessibility above. This next step should increase an easy cooperation of the document with the correctional software. These languages were chosen because of their good expectations.

6.1.1 C++

This language has belonged to programming mainstream recently. Even MS Windows application is created in C++. This language is very useful when more computing power is required and it is suitable for larger applications. But it needs a quite detailed source code. Assembling from the scratch could be wasteful from the view of required time. Even with existing libraries and code sequences the assembling would take more time.

6.1.2 Visual Basic for Applications

It is a version of programming language based on Visual Basic syntax. The language is equipped with the common instructions and commands from Visual Basic. It is improved with a special sort of commands exploiting potential of different host applications (e.g. MS Word, MS Excell …) not only from Microsoft production. VBA is capable of quick idea transcription into a graphic interface.

6.1.3 Python

Python programming language is not a mainstream language but it is still very popular not only among its users. Its source code can be interconnected with other programming languages. The syntax is synoptical and the final source code is as well. Python is usually used for economic in-house systems. The application Vlnka 0.4, mentioned before, has been developed with the Python language.

6.1.4 PHP

Originally it was a script language which complemented dynamic possibilities of static HTML pages. Then in the process of development it became a complex programming language suitable for large applications, although it is used for web pages mostly. It can not get along without a server because its language interpreted on the server side. PHP differs from other languages mentioned before. It can not be functional as a self-standing application because of needed interaction with server and usually some kind of an internet browser. Consequently, thanks to this, it is the ideal solution for online software version.

6.2 Programming language – the choice

I have discussed this languages selection with programmers from addressed companies dealing with software developing (they are going to be mentioned later in the finance section). Surprisingly their opinion corresponds closely with a programming language comparison from [5], on the 4th page. I have chosen potential languages for our use and organized them into the following table. The original short verbal ratings has been transcripted into the numeral values and sorted by their importance. This allowed a numeral classification of applicability. I have modified the data importance with its weight.

Tab. 3 Used rating scale

Excellent	Good	Satisfactory	Unsatisfactory
3	**2**	**1**	**0**

Tab. 4 Rating of some programming languages

	Processing speed	Source assembling speed*	Object orientation*	GUI*	Suitable for large projects	***Applicability***
C	3	0	0	2	2	**9**
C++	3	1	3	2	3	**18**
Pearl	1	3	1	2	1	**14**
Python	1	3	3	2	3	**20**
Visual Basic	2	3	1	3	0	**16**
* The rating was multiplied by two because of items weight.						

Programming languages rating results

The best result is represented by Python with 20 rating points, C++ is the second with 18 points and the third is Visual Basic with 16 points. Although the rating is dealing with Visual Basic, we can take it as VBA because these languages are almost the same. I was recommended by a programmer of Sophics Ltd to use Java language because of its particular advantage. This language was not included in the rating but I have denied this option anyway because of the other important fact.

The use of first two representatives was not recommended for their demands. Here comes a little explanation. The software can be assembled with the use of Python or C++. In that case it would be better to assembly from an existing source codes and programmers are usually equipped with different code fragments with usually demanded functions. Unfortunately our typographic correcting software should be capable of common functions only a little. All other functions and algorithms will have to be programmed manually and this represents quite big amount of source code written line by line. Although the programming in Python and C++ is not a problem (it is popular among programmers and used quite often) I have been told there is another better way.

Visual Basic for Applications – advantages

The VBA use was the mostly recommended application solution. The main reasons are easily given by:

- an utilization of VBA programming included in every MS Word version,
- a guaranteed compatibility with all MS Word supported file formats,
- quick and easy GUI creation with familiar interface.

All these small advantages contribute to shorter assembly time and increase the potential of use. Although the combination of MS Word with VBA could be limiting to the Windows operational system, this is not disadvantageous at all. That is because we would like to hold an average user‘s interest, and MS Windows with the office suit and its components from the same manufacturer is today’s PC equipment standard. Other alternative operational systems as a different Linux distributions or other even a quality and noncommercial office suits are enforcing only gradually.

7 A FORM

I have already determined requirements for the correctional software, the basic typesetting system to cooperate with. And finally a programming language has been chosen in the previous chapter. We are standing at the point, in which we are aware of our working devices. But still there have to be more matters formulated. This seventh chapter is going to describe the subject of our effort. I would like to make clear the way of correctional software system functioning, select all applicable corrections, and illustrate the program use.

Finally there is going to be submitted a financial proposal for the case of commercial development by third-party companies. Let us now take a look at the first point.

7.1 Functional form

When I was thinking about this application, I have tried to figure out the best solution for its functioning. My first steps led to the self-standing application with only a few dialog menus. A goal was to come out with the easiest way of preparing to the functional condition, setting to user's requirements and use. I thought that the only installation file with several dialogs and user menus is the best way. This has been quite a long time before I had revealed Vlnka for OpenOffice.org. Just then I have realized my approach to the question of functionality is based on wrong point of view.

An interconnection of application with its supportive documentation

It is self-evident fact I have to start formulating functional form from pro and cons given by used typesetting systems and programming language. This offers a great chance of program documentation. Any software is usually accompanied with its documentation describing software functions, settings, operating, etc. This is done by different enclosed files (the most used file formats are *.pdf, *.txt,*.hlp, *.html or its compiled version *.chm).

The use of MS Word offers an easy interconnection of text document with an application written in VBA and hidden on the background. This hidden application is saved in the text document source and can be viewed only through the VBA editor that means a common user is not aware of its presence. The application can be activated through a defined shortcut or a button inserted in the document structure whenever the user wants to.

The security question of possible modification or a virus infection is going to be discussed later in the. Let us take a look at the application procedures. In the following figure I have divided the action between a user and an application. The notes in the very right column are explaining more detailed process of actions on the background.

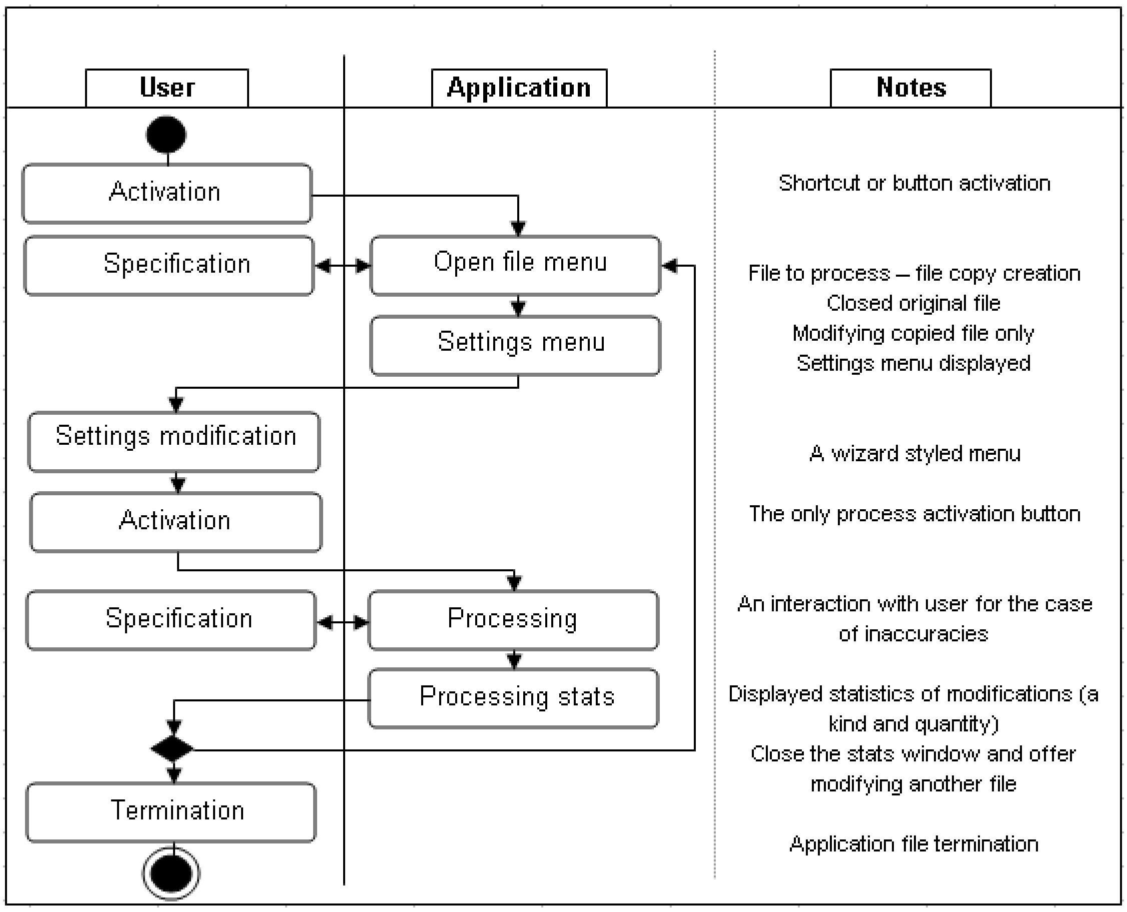
User
Application
Notes
Activation
Specification
Open file menu
Settings menu
Settings modification
Activation
Specification
Processing
Processing stats
Termination
Shortcut or button activation
File to process – file copy creation
Closed original file
Modifying copied file only
Settings menu displayed
A wizard styled menu
The only process activation button
An interaction with user for the case of inaccuracies
Displayed statistics of modifications (a kind and quantity)
Close the stats window and offer modifying another file
Application file termination

Fig. 4. Typographic correctional application process

We cannot think about this application as only a summary of some macros. The most of these software particular processes are brand new to the MS Word. It is going to bring absolutely new sequence of VBA source code, therefore the application development has to be based on programming not on macros aggregation [19].

The concept submitted in previous paragraphs produces a tight connection of program and the documentation. It is high time to explain, that the documentation is actually a reference manual for some typographic rules. At the first place it has to provide an interpretation of some typographical rules and not only those covered by the application. The best solution is represented of well-informed user who is aware of these rules, and is not bound to this program anymore. The second purpose is only conventional program documentation.

7.2 Range of corrections

Although it sounds very nice to have all typographical mistakes corrigible, it is almost an impossible task. Some mistakes are easy to find with simple elementary character analysis. Those algorithms or computational procedure are relatively easy programmable. But there are a large number of corrections that can not be done without the document context. This kind of remedying requires user's interaction then. This interaction has to be only as the most necessary as possible. The program asking for the same question sixty times, this can happen with voluminous document or for example with large included data tables etc., is bothering the user. Any annoying application has the best expectation of being disused successfully more then one time, especially by the common user.

Therefore these cases for correction have to be chosen with a care. The application for document corrections was originally intended for covering the most possible quantity of typographical errors. This plan was changed later because of a complicated realization. Some errors can be discovered only thanks to a connection with some explicit violation of a certain kind.

As noted above, contemporary typesetting systems are equipped with a spell checking and eventually a useable grammatics checking too. It means that any written word is going to be checked for misspelling and even a wrong accidence. But there are some typographic rules about diacritical marks and specific characters in the Czech language. Even a space between the words is a character that is directed by typographic rules in many different ways. Surprisingly this very character contains the most typographic errors at itself. These errors come from the straight matter rules.

The rule of writing different kind of spaces affects these easily recognizable cases:

- diacritical marks (missing or redundant space),
- prepositions (used breakable space where unbreakable is recommended),
- a hyphen or en-dash (a confusion of the meaning and the difference),
- an abbreviation and academic degree (missing, redundant or wrong space inserted),
- numbers (separating a group of large numbers by unbreakable space),
- numbers with units and their multiples (missing or wrong space inserted).

Although the typography discerns really quite a lot different kinds of typographical spaces in typesetting, we can talk about two fundamental kinds in use with MS Word; breakable and unbreakable space. This approach is the most similar with the common user‘s one.

There are of course a lot of other rules, but those noted above in the enumeration are the easiest from the point of detection. Further typographical rules can be found well-arranged in [1] and [10].

The fact of unfamiliarity with basic typographical rules pointed out the need of additional education. This goes luckily towards our supposed application form – the document with the hidden application. Next chapter is going to explain the role of the document and its content.

7.3 Supportive document

Some documents, usually enclosed as an application‘s documentation, has mostly one simple task. It is informing an application’s user about software capabilities and its managing. In this case the use of correctional program can be explained on one single page of size smaller than standard paper size A4. This page contains of a few points:

- a very brief program introduction (can be less than 4 lines of text),
- a short and structurized operation manual (explained following general steps after an activation button is pressed, synoptic and brief information in short sentences),
- a cross-reference to additional information on other pages (those pages have to contain the basic Czech language rules connected with correct using of different spaces and other frequently used characters),
- the button activating a correctional application.

Other pages are intended for the explanation of those frequently used and, therefore, violated rules.

I have tried to put together some of those rules, and I have made of them a manual which is enclosed as the P II in appendices section. With the final program this manual should be given in two different versions with the short and the full length one. This arrangement is going to meet different levels of user’s interest. That same information is going to be given on the supportive web pages (more information in chapter 8.2).

7.4 Custom-made correctional software

We can text compare those ways of commercial and noncommercial software development, in other words a financial side of a different decisions. First of all let us take a look at the common commercial correction (or proof-sheet) charges.

Estimated charges of document correction

Following information comes from a Plow & Piston, Ltd and its web page presentation[3]. This company is specialized in documents correcting not only in Czech language. They provide a wide variation of different correcting, proof-sheet and translation services.

Let us imagine an easy task – the complete document check (of grammatics, stylistics, and typographical errors), no time limitation, highlighted changes done in the text and a correction statistics. The document is made of 75,000 characters, i.e. 42 standard pages of text by valid company‘s price list.

Tab. 5 Price estimation of document correction by Plow & Piston Ltd charges

Item description/a figure/details		
Spell check, grammatics and stylistics check, typographical validity (DOC, RTF, HTML files)	**130 Kč**	per 1,800 characters
Highlighting the corrected part	**10 Kč**	additional charge
Deferred correction (no time limit)	15%	reduction of final price
Number of pages	42 pcs.	i.e. 75,000 characters
Estimated final price after reduction	**4,998 Kč**	final price

By the official figures in the table above this correction service for one document amounts to 5.000 Kč. This value can be thought as an equivalent costs to those connected with the way of self-correction done by a help of accessible software tools. This correction price is not typical. Company Plow & Piston belongs to a higher standard of provided services, its charges comes up to this fact. On this level a quality of corrected documents is guaranteed, which is not sure with another solution with lower price.

[3] http://www.korektura.cz

Estimated price of custom-made correctional software

I have tried to figure out more accurate financial results, but the approach of software developing companies to their price estimation is really very cautious. Instead of precise estimation we can only guess the amount of needed financial resources; even we have discussed basic conditions, required functions, its GUI and functioning fundamentals.

There are some crucial factors affecting the final costs which were the most problem with the price estimation. Let us mention just a few of them: programming language, an application size caused by amount of unpredictable growing correction conditions, updates and user customization, etc. By this work previous conclusion we have formulated following figures in compliance with usually used programmer's pay.

Tab. 6 Common wages according to the programmer's level

Programmer's level	**price**
Application programmer	200 — 1000 Kč per hour
Brigade-programmer (e.g. student)	80 — 130 Kč per hour

The final price is then counted as a simple sum of wages and additional costs modified by the possible discounts (depends on contractual conditions). Fortunately this software is not so much time-consuming. Its need of assemblage time was estimated to approximately one or two weeks, financially i.e. 20,000 Kč (250 Kč per hour, 8 hours a day, 10 days).

This solution is only partial, custom-made software is going to cover only typographical corrections (and only a part of them). The use of different software with different aim is needed then. But still, this lowers a dependence on another subjects, and therefore another costs. Document correcting is only limited to the used software and its capabilities.

Estimated use of only internal sources of TBU

This is surprisingly the most controversial part because of the costs estimation. The price here is not expressed by money value. It is represented by the number of time spended on official discussions, theoretical preparations, and an effort expended by particular voluntary programmers. Perhaps these programmers could be easily found in university academic area, because a continuity on this work is an ideal theme for someone's master thesis. In this case almost any money does not have to be spended and TBU can take advantage of another new possible way of propagation (inserted university logo and credits).

8 DISTRIBUTION AND USE

This chapter contains other information relevant to the correction software but from another point of view. This information is supplemental and they extend the understanding of this work finding.

8.1 Application benefit

It is useless explaining that time is money in our modern world. Please, allow me to say another one self-evidence. A man is inventing machines to make his work faster, easier and better. This correctional application‘s aim is not very perceptible for a common user and his first sight. It is going to do something, he has never needed. It may seem as unnecessary thing. But there are many people dealing with the typography everyday.

This application benefit can be measured mathematically. There exists work measurement method called AdminMOST. It is a variation of the MOST® (The Maynard Operation Sequence Technique) a measurement system developed from analyzing basic work operations and its elements. This method can specify the amount of TMU (time measurement units) needed for a particular operation; in this case it is the manual correction of typographical errors in document. With no hesitation I can say, this calculation would not bring any satisfactory result. Easily the numbers of unknown typographical errors at any location in, for example 75,000 characters, document can only bring a very high number. This analysis can be accurate, but its result has no helpful usage, because it has been already known.

The real benefit can be only represented by saved correctional time (i.e. the difference between automatic and manual processing). It is a few minutes for one short document and one person or a few seconds for software. It can be also an hour for a large document or just a few seconds for software. In the case of batch processed correcting is a comparison nearly redundant, but it is still only those few seconds for software…

There are more then 10,000 students at Tomas Bata University this year. Many of them are going to write some documents for their academic tutors, and probably their bachelor and master thesis too. Even this figure could be higher more than, let us say 10 %, it is still one thousand documents then. These documents should be written correctly. Imagine those saved seconds and much better those minutes and multiply this with one thousand documents. The result comes to the hundreds of hours. This number is strongly underestimated. We can add the saved time of those possible users from the next paragraph. It may seem unreal, if you think so, it is better not to count and just accept that.

8.2 Targeted market

There are not only our university students, who can take advantage of this product. There are redactors, editors, book publishers, web presentation creators and administrators, teachers, commercial advertisement creators, and many others advanced users with the need of typographical rules in their daily life. Although this application is very specialized, it can find an amount or regular users.

8.3 Application security

There are three categories of user's awareness of VBA macros in MS Word. The first category has never heard about it. The second one thinks that macros in MS Word are just dangerous things useful only for infecting computer with threatening viruses. The third category uses VBA macros. For the file security can be used these following precautions. They are quite simple but effective.

The simplest precaution is the basic security necessity for this application too (i.e. document). The document (*.doc) has to be read-only under the password protection. This option locks all modifications of document and VBA source, although this is only basic level, it offers an easy way of file protection for the user and for the application too.

Another basic security precaution is connected with a MD5 file checksums. The original file can be packed (for better internet data transfer) and its unique checksum can be created. In the case of possible file modification this checksum is different. Official checksum file is going to be accessible on supportive web pages (next chapter). This checksum should be made for each officially released application version. This security is not absolute and final, but it even offers an affirmation that the file had not been changed by unauthorized person.

These were only some simple and quickly applicable precautions.

8.4 Supportive web pages

It is better for some kind of projects to be accompanied by supportive web pages today. This is undoubtedly that case. This web presentation should be multilingual (at least Czech and English version) and it should include:

- brief information about application purpose, actual version and features,
- simple instruction manual and explained security features,
- a full version of support documentation (i.e. basic typographic rules),
- download and checksum section,
- users discussion,
- some other features (on-line correcting…).

8.5 Possible on-line correcting

This could be an additional service accessible by supportive web pages. This feature can take two forms. It can provide a common service as this work object or it can modify inserted HTML source code with an unbreakable space where needed. This application can be easily programmed with PHP; its capabilities can be modified only to those very simple typographical rules. The PHP source has a small disadvantage – it can be copied from a server and used somewhere else. I was recommended to thing about this application written in Java and then compiled. This application can run invisible on the server background. Anyhow, this service would increase rapidly time needed for complete realization, therefore it has only marginal importance. The main part of this paper work is a presentation and explanation of the application for typographical errors correction written in VBA.

CONCLUSION

The paper work clearly indicates that this software construction under Tomas Bata University patronage offers a potential of possible realization and successful usage. All sources I have explored and summarized prove that there is an uncovered segment of present electronic document correcting. Unfortunately this segment is quite small not only for software developing companies but also absolutely insignificant for that giant representative of Microsoft Company.

Our Czech language is so powerful with all its expression spheres and rich accidence. Surprisingly it is not always a good feature. Especially the written form of Czech language disposes of many different rules and the aim of the correctional software is to cover those dark places of user's nescience. This work showed that commonly used DTP systems with the massive majority of Microsoft Word are able of offering different Czech language correctional tools but some are still missing.

Presently used correctional equipment covers spell checking and even advanced grammar check. As I have discovered the knowledge of basic typographical rules and their usage is of course somehow supposed. But that kind of typographic errors is not generated only by the lack of user's education. It can occur even from the simple coincidence caused by inaccurate data insertion or during data modification.

It is quite easy to explain why this work could be something useful and handy. There are very good reasons at the both sides for creating correction software, not only for student's thesis. The first side is represented by students themselves. It might be is their first time or may not. But they have to take care of many things for sure. Formal lay-out and set-up is important. But not as important as the very content of their work. Amount of rules, precepts and thoughts mixed up together is making their work progress slow and less efficient. The second side is held by their teachers and tutors. Let us be honest, they should read each student‘s thesis themselves. The more the thesis is written incorrectly, the more its reader has to make an effort.

There are more then 10,000 students at Tomas Bata University this year. I think the point is now clear. But those potential users can not be limited only on those students attending Tomas Bata University and a part of personnel. In the case this application is offered freely on the internet web pages for public use, the amount of potential users grows nationwide. I think this application can find another number of grateful users among foreign student who are going to write their thesis in Czech.

This correctional software can be developed with the use of only internal university sources. I am sure there are some students interested in programming with needed acquirements and thanks to the newly established Faculty of Applied Informatics is software engineering very close to the university competency and its possibilities. The investigative character of this work makes itself an ideal theme for the master thesis. As a result this application can be then fully used for propagation of Tomas Bata University.

It is possible that this software development will not find its support from the Tomas Bata University. In that case I am going to offer this work results to the interested person or parties from the public. I hope this could help someone.

During my reconnoitering, I have realized the quantity of many different rules given by our national language. Many of them I have met for the first time, even I was using them before – mostly incorrectly. Since then, my awareness of correct Czech language has improved. Surprisingly the most influence had those simple basic rules. It would be great to offer these helpful materials to the others not only students too, just for a case…

The typography is about those visible things that have to be seen. The purpose of all typography objects is to be read or viewed, and finally even remembered. The aesthetics and readability comes hand-in-hand.

BIBLIOGRAPHY

[1] BERAN, Vladimír, et al. *Typografický manuál : aktualizovaný*. 2. upr. vyd. Praha : Kafka design, 2000. 160 s., 1 CD-ROM.

[2] BRÁZDA, Jiří. *PHP 5 : začínáme programovat*. 1. vyd. Praha : Grada Publishing, a.s., 2005. 244 s. ISBN 80-247-1146-X.

[3] ČERNÝ, Jaroslav. *Excel 2000, 2002, 2003*. 1. vyd. Praha : Grada Publishing, 2004. 192 s. ISBN 80-247-0923-6.

[4] HALVORSON, Michael. *Visual Basic 5 : krok za krokem*. 1. vyd. Brno : Computer Press, 1997. 370 s. ISBN 80-7226-036-7.

[5] HARMS, Daryl, MCDONALD, Kenneth. *Začínáme programovat v jazyce Python*. 1. vyd. Praha : Computer Press, 2003. 451 s. ISBN 80-7226-799-X.

[6] HLAVSA, Zdeněk, et al. *Pravidla českého pravopisu*. Redaktorka publikace Jitka Spoustová. 2. vyd. (s Dodatkem Ministerstva školství, mládeže a tělovýchovy České republiky), Praha : FORTUNA, 2005. 391 s. ISBN 80-200-1327-X.

[7] HRADSKÝ, Josef, et al. *Český Office 2000 Professional*. 1. vyd. Praha : Grada Publishing, 1999. 476 s. ISBN 80-7169-882-2.

[8] HLAVENKA, Jiří. *QuarkXPress : podrobná uživatelská příručka*. 1. vyd. Praha : Computer Press, 1998. 260 s. ISBN 80-85896-09-5.

[9] KISZKA, Bogdan. *1001 tipů a triků pro programování v jazyce Java*. 1. vyd. Pra-ha : Computer Press, 2003. 514 s. ISBN 80-7226-989-5.

[10] KOČIČKA, Pavel, BLAŽEK, Filip. *Praktická typografie*. 1. vyd. Praha : Computer Press, 2000. 289 s., 1 CD-ROM. ISBN 80-7226-385-4.

[11] KOPKA, Helmut. Latex : podrobný průvodce. 1. vyd. Brno : Computer Press, 2004. vii, 576 s. ISBN 80-722-6973-9.

[12] MCCARTHY, Jim. *Softwarové projekty*. Praha : Computer Press, 1995. 190 s. ISBN 80-7226-164-0.

[13] MĚCHURA, Michal Boleslav. *Vyšší škola Wordu*. 1. vyd. Brno : Nakladatelství Šimon Ryšavý, 1996. 279 s. ISBN 80-902143-1-2.

[14] MORKES, David. *Učebnice Visual Basicu 6.0*. 1. vyd. Praha : Computer Press, 2000. 176 s. ISBN 80-7226-312-9.

[15] PALETA, Petr. *Co programátory ve škole neučí : Softwarové inženýrství v reálné praxi*. 1. vyd. Praha : Computer Press, 2003. 331 s. ISBN 80-251-0073-1.

[16] ROMAN, Steven. *Programujeme makra ve Wordu*. 1. vyd. Praha : Computer Press, 2000. xx, 357. ISBN 80-7226-273-4

[17] RYBIČKA, Jiří. *LaTeX pro začátečníky*. 3. vyd. Brno : Konvoj, 2003. 238 s. ISBN 80-7302-049-1.

[18] ŘÍHA, Petr. *Slovník počítačové informatiky*. 1. vyd. Ostrava : Montanex, 2002. 261 s. ISBN 80-7225-083-3.

[19] SCOTT, D.F. *Velká kniha programování v Microsoft Office 2000*. 1. vyd. Praha : Computer Press, 2000. 938 s., 1 CD-ROM. ISBN 80-7226-240-8.

[20] ŠIMEK, Tomáš, VACEK, Jiří. *Microsoft Word*. 1. vyd. Praha : Computer Press, 1999. 190 s. ISBN 80-7226-216-5.

[21] ZANDIN, K.B. *MOST : work measurement systems*. 3rd rev. and expanded edition. New York : Marcel Dekker, 2003. xxiv, 519 s. ISBN 0-8247-0953-5.

OTHER RESOURCES

[22] Typography – Wikipedia, the free encyclopedia [online]. 2007, [cit. 2007-03-12]. <http://en.wikipedia.org/wiki/Typography>

[23] ŘÍHA, Jozef. *Tvorba elektronických šablon pro UTB ve Zlíně*. Zlín, 2005. 86 s. Univerzita Tomáše Bati ve Zlíně. Vedoucí diplomové práce Ing. Pavel Stříž.

[24] CHIP. 2005, roč. 15, č. 12. Praha: Vogel Burda Communications s.r.o., 2005. 1x měsíčně. ISSN 1210-0684.

[25] ČSN 01 6910. Úprava písemností zpracovaných textovými editory nebo psaných strojem. Praha : Český normalizační institut, 2002. 34 s.

[26] ČSN ISO 1000. Jednotky SI a doporučení pro užívání jejich násobků a pro užívání některých dalších jednotek. Praha : Český normalizační institut, 1997.

[27] Microsoft Office – Wikipedia, the free encyclopedia [online]. 2007, [cit. 2007-03-12]. <http://en.wikipedia.org/wiki/Ms_office>

[28] OpenOffice.org: home [online]. 2007, [cit. 2007-06-04]. < http://www.openoffice.org/>

[29] QuarkXPress – Wikipedia, the free encyclopedia [online]. 2007, [cit. 2007-04-05]. <http://en.wikipedia.org/wiki/QuarkXPress>

LIST OF ABBREVIATIONS

DTP	Desktop Publishing
GUI	Graphical User Interface
HTML	Hypertext Markup Language
MD5	Encryption Algorithm
MS	Microsoft
OO.org	OpenOffice.org
PC	Personal Computer
PHP	Personal Home Page Tools or Hypertext Preprocessor
TBU	Tomas Bata University
TMU	Time Measurement Unit
VBA	Visual Basic For Applications
WWW	World Wide Web
WYSIWYG	What You See Is What You Get

LIST OF FIGURES

LIST OF TABLES

APPENDICES

APPENTIX P I: THE LIMITATION OF ENGLISH SPELL CHECKING

Eye halve a spelling checker
It came with my pea sea
It plainly marques four my revue
Miss steaks eye kin knot sea.

Eye strike a key and type a word
And weight four it two say
Weather eye am wrong oar write
It shows me strait a weigh.

As soon as a mist ache is maid
It nose bee fore two long
And eye can put the error rite
Its rare lea ever wrong.

Eye have run this poem threw it
I am shore your pleased two no
Its letter perfect awl the weigh
My checker tolled me sew.

This poem is modified version of original Jerrold H. Zar's version. A spell checker would not find any mistakes here because all words are written correctly, even thought there are many of them.

Sources

This version is accessible from http://www.buzzin.net/english/spellchk.htm.

Full length version is accessible from http://www.jir.com/pullet.html

or

The Journal of Irreproducible Results, vol. 39, #1, January/February 1994, page 13, and vol. 45, # 5-6, 2000, page 20.

APPENTIX P II:

Basic typographical regulations of unbreakable space usage

(Adapted to MS Word)

Unbreakable space – such a kind of space, not allowing its placement at the end of a line ***(Ctrl + Shift + Space).***

About spaces generally – there has to be only one space between two words (two or more consecutive spaces are considered as error), we can discern two different kind of space in MS Word, a common and unbreakable one.

Punctuation symbols

- all **punctuation symbols** have to be placed instantly behind a word
- there has to be **space behind every punctuation symbol** (the only exception is a symbol as a part of numeral statement)
- space ahead of ellipsis points ***(Alt + 0133)***

Prepositions and conjunctions

- there have to be no one character prepositions at the end of the line**:** "***s*, *S*, *v*, *V*, *z*, *Z*, *k*, *K*, *u*, *U*, *o*, *O*** and conjunctions ***A*, *i* and *I***"
- unbreakable space ahead of character "**a**" (It is better logically and optically to have this conjunction at the end of a line.)
- unbreakable space ahead of two characters prepositions is recommended (od, ke, do …)

Symbols

- "**&**" ampersand has spaces on both sides, the space behind it is unbreakable
- "+" and "=" have spaces on both sides
- "/" a slash is written without spaces
- "§" has to be followed by unbreakable space and a number

Hyphen and en-dash

- **hyphen** – splits words, is written without spaces ***(Alt + 0173)***
- **dash** – expresses sentential pause or range, unbreakable space ahead of it ***(Alt + 0150)***

Parenthesis

- there has to be **space ahead of opening parenthesis**
- **no space behind opening or ahead of closing parenthesis**
  (text text) – correct, (text) – wrong!

Quotation marks

- is written without spaces **(„text text" – correct, „ text text " – wrong)!**
- MS Word can automatically replace "English" quotation marks with „Czech" ones
  Tools > Automatic corrections > Automatic format correction

Numerals

- date and time – 26. 6. 2003; 26. června 2003; 10.45; 44:50,25
- large numbers are separated by unbreakable space at the count of three numbers, the meaning of decimal point and decimal comma is switched
- **postcode** is always written **in the** "3 + 2" **scheme** (760 05)
- **multiple is written by a number without space and** "**x**" (5x, 124x, …)

Numerals with specified unit

- numeral followed by unit is **separated by unbreakable space** (10 km)
- **numeral followed by unit but not separated by space express an adjective!** (9m jump)
- geographic coordinates are without spaces (52° 30' 20")
- international abbreviation of currency and number are separated by space (CZE, USD, …)

Abbreviations

- **unbreakable spaces** (s. p.; a. s.; př. n. l.; t. r.; s. s r. o.; s. r. o.; v. o. s. …)
- company name followed by its type is **always separated by a comma** (Alfa, a. s.),
  in the case **company type is written ahead of its name there is no comma** (a. s. Alfa)

- a period is not written behind**:**
  1. fa – firma (faktura), pí – paní, circa – cca, etc.
  2. **International measurement units** SI (Systéme International d'Unités),
     ČSN ISO 1000 since 1997

  3. ČR, ČD, OSN, VZP, … – abbreviations written with uppercase characters

Academic degrees

- **CSc., DrSc, MBA, Ph. D., Th. D., DiS. is written ahead of surname**, it is separated by a comma in between (Ing. Petr Hanuš, CSc.)
- pedagogic degree is written with small first character (**prof.**, **doc.**, **as.**, **odb. as.)**
- **academic degrees are defined in:** ČSN 01 6910, Zákon o vysokých školách,

Further information can be found in [1], [6], [10], [20], [25], and [26].

APPENTIX P III:

Základní typografická pravidla o použití pevných mezer (Czech version)

(Přizpůsobeno pro prostředí MS Word)

Pevná mezera – je takový druh mezery, který nepovolí tuto mezeru na konci řádku. ***(Ctrl + Shift + Space)***

Mezery obecně – mezi slova patří jen jedna mezera (dvě a více za sebou jsou chyba), v prostředí MS Word rozlišujeme běžnou a pevnou mezeru!

Interpunkční znaménka

- všechna **interpunkční znaménka** sázíme vždy těsně za slovem
- po interpunkčním znaménku **musí být mezera** (výjimkou je znaménko jako součást číselného vyjádření)
- před trojtečkou je mezera ***(Alt + 0133)***

Předložky a spojky

- Na konci řádku nesmí být předložky**: *s*, *S*, *v*, *V*, *z*, *Z*, *k*, *K*, *u*, *U*, *o*, *O*** a spojky ***A*, *i* a *I***
- před **a** je pevná mezera (raději než na konci, opticky a logicky je to lepší)
- vyhnout se dělení textu i za předložkami z dvou písmen (od, ke, do, …)

Symboly

- & mezery z obou stran, za ním pevná
- + a = jsou na obou stranách odděleny mezerami
- / se píše bez mezer
- po § následuje pevná mezera a číslo

Spojovník a pomlčka

- **spojovník** – rozděluje slova, píše se bez mezer ***(Alt + 0173)***
- **pomlčka** – je větná pomlčka nebo interval, před pomlčkou je pevná mezera ***(Alt + 0150)***

Závorky

- před otervírací závorkou má být mezera
- **nikdy nepíšeme mezeru za otvírací závorkou nebo před zavírací (text text) – správně, (text) – špatně!**

Uvozovky

- píší se **bez mezer („text text" – správně, „ text text " – špatně)!**
- MS Word disponuje automatickým nahrazováním "anglických" uvozovek „českými"
  Nástroje > Možnosti aut.oprav > Automatické opravy formátování při psaní

Číslovky

- datum a čas – 26. 6. 2003; 26. června 2003; 10.45; 44:50,25
- velké číslice se oddělují **po třech** od čárky **pevnou mezerou** na obě strany
- **PSČ** se píše ve tvaru 3+2
- **násobek** se píše **bez mezery** (5x, 124x, …)

Vyjádření jednotek

- mezi číslem udávajícím míru a jednotkou je **pevná mezera** (např. 10 km)
- pokud **mezera není**, jedná se o **vyjádření přídavného jména!** (např. 12% roztok)
- **úhlové stupně, minuty a vteřiny** se píší **bez mezer**
- měna vyjádřená mezinárodní zkratkou **(CZE, USD, …) má mezi sebou a číslem mezeru**

Psaní zkratek

- **zkratky sousloví** obsahují pevné mezery! (např.: s. p.; a. s.; př. n. l.; t. r.; s. s r. o.; s. r. o.; v. o. s.)
- Právní forma firmy **zkratkou za názvem se vždy odděluje** čárkou: Alfa, a. s.,
  ale je-li právní forma **před názvem**, tak se čárka **nepíše**: a. s. Koramex; v. o. s. Klasik.

- Tečka za zkratkou **se nepíše** v případech**:**
  4. např. fa – firma (faktura), pí – paní, circa – cca aj. – píše se bez tečky
  5. **mezinárodní soustavy jednotek** SI (Systéme International d'Unités),
     ČSN ISO 1000 z 1997

  6. ČR, ČD, OSN, VZP, … – zkratky velkými písmeny

Psaní akademických titulů

- **CSc., DrSc, MBA, Ph. D., Th. D., DiS.** píšeme **za příjmení** osoby, **mezi příjmením** a **zkratkou** píšeme vždy **čárku**
- zkratky učitelských titulů **před jméno** ***malým počátečním*** písmenem:
  prof. - profesor, **doc.** - docent, **as.** - asistent, **odb. as.** - odborný asistent.

- **Zkratky vysokoškolských a vědeckých titulů** uvádí ČSN 01 6910, Zákon o vysokých školách a Pravidla českého pravopisu

Další informace obsahují [1], [6], [10], [20], [25] a [26].