Do You Speak and Write in Informatics?

Maria Csernoch
Faculty of Informatics, University of Debrecen
Debrecen, Hungary

ABSTRACT

End-user text management, the handling of digital text-based documents by non-professional end-users, is one of the most frequent and contradictory computer related activities. Most of these documents are loaded with errors, which originate both from the end-users' lack of knowledge in the subject and by their ineffective problem-solving approaches. The aims of the paper are to clarify the theoretical background required to handle digital texts, to provide a definition of a correctly formatted and edited text, and to present a classification of errors. Based on error-related research, studies, and approaches in programming and spreadsheets, we define the classes of qualitative and quantitative errors and their sub-classes in digital texts. We further claim that end-user text management requires proper training and, in everyday usage, similar to other sciences and subjects, concept-based problem-solving approaches supported by firm reliable schemata in the background.

Keywords: digital texts, errors, error-recognition, concept-based problem-solving, TPCK

1. INTRODUCTION

Most of the creators – the authors and modifiers – of text-based digital content are faced with several challenges and problems which they are not prepared for and which they cannot solve effectively and efficiently [9][11][12][2][3][32]. One of these challenges, which is the primary concern of the present paper, is the understanding and handling of the semi-artificial language of text-based digital documents (referred to as d-documents) of different forms, such as digital texts, presentations, webpages, spreadsheets, etc., and the interfaces to handle them.

2. THE SEMI-ARTIFICIAL LANGUAGE OF D-TEXTS

The faces of printable characters

It is obvious that text-based contents are written in natural language(s). Usually, the language of the text is homogeneous, but pieces of texts can be inserted in languages other than the dominant language [10]. However, the different character sets combined in a handwritten text never require the accuracy of machine-printed documents. This leads to one of the paradoxes of d-documents: in d-documents, compared to handwritten texts, creators do not have to form characters, but must be aware and more conscious of the faces of characters, especially foreign language and special characters [5]. In machine-printed outputs, regardless of the interface, the face of the character is clarified, and so cannot be hidden behind negligent handwriting. Consequently, if incorrect characters are typed and/or the number of characters does not match the requirements of the language, this leads to syntactic, and in some cases, to semantic errors.

Non-Printable and/or Separator Characters

The other concern of d-documents is handling characters which do not exist in handwritten texts. The most frequently used separator characters in d-documents are the space (U+20), nonbreaking space (no-break-space, U+A0), end-of-paragraph (enter) (U+A), and tabulator (U+9), characters [43][44][45][46]. These characters are non-printable characters – in most cases it is felt they are annoying and better unseen – but they are part of the documents and play a crucial role in text editing. However, hiding these characters in d-documents makes end-user text management even more difficult, because non-trained creators work blind, and consequently, are exposed to more challenging tasks and problems.

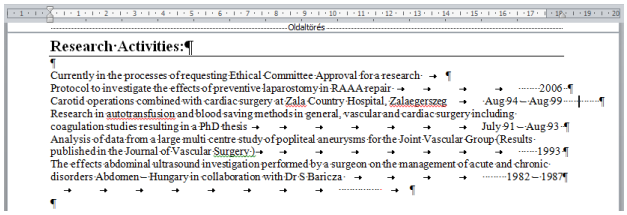


Figure 1. Improper use of non-printable characters (space, enter, tabulator) [private collection].

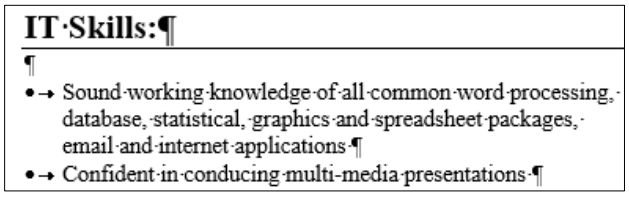


Figure 2. Statements from the overconfident author of Figure 1.

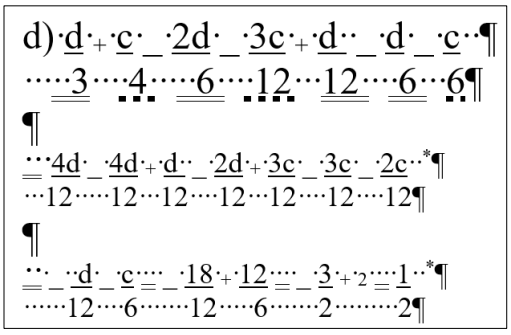


Figure 3. An example of high-level bricolage from a preservice teacher of mathematics and informatics [private collection].

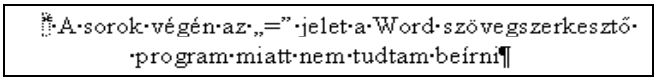


Figure 4. An overconfidence preservice teacher of mathematics stated that 'Because of the Word word-processor, I was not able to insert an "=" sign at the end of the line' (Figure 3). In general, the statement is true, because Word does not underline any space characters at the end of lines, which tool she used to mimic the = sign (double underlined) [private collection].

The consequences of the improper usage of non-printable characters leads to bricolage [9][2][3], and as such, lowers the effectiveness and efficiency of document management [18]. The improper use of non-printable characters – unnecessary space, enter, and tabulator characters – do not fulfill the requirements of a properly formatted and edited text (Section 5) (Figure 1, 3, 5, 7).

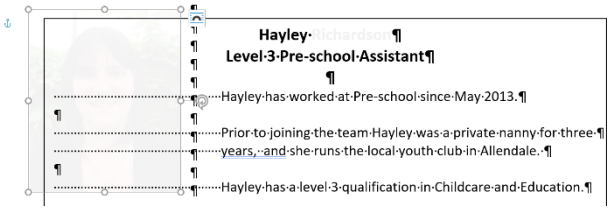


Figure 5. Errors from the Parent Handbook of a pre-school [50]. The errors clearly indicate that the teachers are not aware of how to use non-printable characters (space and enter), how to wrap a picture, and how to read and interpret visually presented data; in general, what is correct and what is not in this d-document considering operational use.

Spell Checkers

In the present context, we cannot leave the spell checker programs unremarked. These are the programs which word processors are usually are equipped with. However, we must be aware of their availability and limitations. Regarding syntactic and semantic errors, spell checkers provide some help, but users must be aware that they are not 100% reliable. Despite the fact that spell checkers can 'overlook' errors, their use is highly recommended, since they can find errors which creators, especially authors, would miss. However, to take advantage of spell checkers, the language setting should be correct. An improperly set language, depending on the available language set, either marks words incorrectly or ignores everything, depending on whether the language is installed or not, respectively.

3. APPROACHES TO ERRORS

To err is human – to debug is a computer skill

To err is human [41]. Consequently, handling errors and misconceptions should be an essential part of the teaching-learning process and of everyday end-user text-management. Since computer sciences and informatics (CSI) is a newcomer compared to other sciences, we can adapt already established methods which suit our interests the best. To teach with errors is a common practice in programming, dating back as far as the '70s [23][24][29]. However, we have found that there is a gap between different computer problem-solving activities: what is everyday practice in programming is not adapted to end-user computing [22][27][26]. On the one hand, end-users do not understand why programmers are so keen on errors. On the other hand, end-user computing is "invisible to IT professionals, corporate managers, and information systems (IS) researchers" [37]. Consequently, there is only a small number of researchers who consider end-user computing and errors, and they are crying out for recognition. Most of the end-users – unfortunately, programmers and other IT professionals in end-user roles are no better than anyone else: they do not know what they do not know, are ignorant of their ignorance [33], and do not care that they do not know.

What makes the situation even worse, is that several commentators consider data and information management 'low level routine knowledge' [1] and even blame it for failures in teaching computer sciences and informatics in schools [25], recommending that it should be banished and be replaced by the teaching of 'serious' programming. However, we claim that end-user activities can be as good as programming for developing the creator's computational thinking [47][14][15][16]. To reach this aim we must find the right proportion of fast and slow thinking [30] which is highly affected by the problem solving approaches applied [15][17].

So, the question is what could explain the high percentage of error-prone uses of spreadsheets [36][38][4][21] and text based documents (word processed texts [2][3][9][11][12][13], presentations [42], web pages [19], etc.).

As was mentioned, the explanation lies in the special circumstance of these software programs and documents. They can be located somewhere between natural and artificial languages. Considering content, handling d-documents is an aspect of natural language processing; however, in terms of the participants in the interaction, they are computer related activities. In spreadsheets we can find traces of handling errors and teaching with errors [36][38][4], which might be due to the programming aspect of spreadsheets and the functional language in the background. However, in other computer related activities errors are not considered as teaching tools or materials.

Approaches to Errors: In the Teaching-Learning Process

Beyond the different approaches to errors, there are several explanations for the avoidance of errors in the teaching-learning process. Francis et al. state that avoiding errors is primarily due to teachers' background knowledge or the lack of it: "…teachers' beliefs were the most influential factor in teachers' intended reactions in situations where students made errors, while the quality of teachers' responses to these situations were determined by their knowledge about the relevant content." [8]. If teachers do not know that there are errors, they do not teach them and are not able to draw the students' attention to avoiding and handling them (Figure 3, 5). Beyond this, teachers' beliefs in an "incremental" the nature of science [6] are as important as their error-handling strategies [15].

Classification of Spreadsheet Errors

Research into the taxonomy of qualitative spreadsheet errors has finally led us to create a similar taxonomy in end-user text management. Quantitative and qualitative errors in spreadsheets have been long discussed [40][34]; however, Leon et al. [34] claimed that quantitative errors are quite well documented, compared to qualitative errors. "Quantitative errors are identified as immediate incorrect numerical values or logic in the spreadsheet, while qualitative errors are associated with spreadsheet design flaws that increase the likelihood of an eventual quantitative error occurring during operational use of the spreadsheet.". The authors go further and claim that "To fully assess the quality of a spreadsheet model and certify it for operational use, it is therefore necessary to identify the presence of both quantitative and qualitative errors in the model." [40].

In this approach we must call attention to the concept of operational use, which makes the fundamental difference between flexible and static documents. The higher the reusability of a document, the higher its operational use, where the reusability of hard-coded documents is zero (Figure 1, 3, 5, 7).

Based on the results achieved in spreadsheet error management, we can formulate our hypotheses to create a similar taxonomy for text-based documents and various end-user activities.

4. HYPOTHESES

[H1] There is a need for a definition of correctly formatted, edited text.
[H2] There is a need for the general acceptance of correctly formatted, edited text.
[H3] There is a definition of correctly formatted, edited text.

5. CORRECTLY FORMATTED TEXT

Based on results from other error-handling research, it is obvious that end-user text management is also in great need of error recognition strategies, considering not only the natural language of the texts, but all the aspects which influence both the quality of the printout and the operational use of the d-document.

How do you know your d-document is right?

As was mentioned in the previous sections, text management is a transition between programming in artificial languages and communication in natural languages; consequently, there are several requirements which a correctly formatted d-document should fulfil. Ignoring and not knowing these requirements leads to errors of different levels in d-documents, which further leads to ineffectiveness, causing serious financial losses in terms of both human and machine resources.

Definition

A d-document is correctly formatted and edited if it is invariant to modification [11] and fulfills all the requirements of a natural language text.

Consequences

- According to the definition, a d-document is correctly formatted, if there is the possibility of high operational use in the document.
- This definition allows the modification of the text – for its reusability –, which is one of the main characteristics of a d-document.
- However, only those modifications are allowed which conform to the user's intentions [11]: beyond the user's original intention, typing and/or formatting are not acceptable [11].
- Beyond this requirement, the text should fulfill the syntactic, semantic, and typographic requirements of the text.

Quantitative and Qualitative Errors

Considering the large number of possible sources of errors in d-documents, we have found that a classification of these errors would be in the users' interest. Our attempt and the similar results found in the spreadsheet environment led us to create a taxonomy of text-management errors. In text-related environments we can define both quantitative and qualitative errors (Figure 6) based on the definition accepted in spreadsheet management [36][40][34].

- **Quantitative errors** result in immediate incorrect grammar, content, or typography.
- **Qualitative errors** are referred to as dormant, or stealth, or latent errors and do not necessarily result in an immediately incorrect text.

In practice, quantitative errors affect and can be detected in the printout. (In this context, we refer to printability in its wider sense, which means that any surface can be considered as a possible output surface, with its specific physical characteristics.) Qualitative errors are latent errors [41], and in text management are usually dealt with when modification is applied to the text.

Classification of Errors

Within the classes of quantitative and qualitative errors further subclasses can be defined (Figure 6). Syntactic (expression adapted from programming), semantic, and typographic errors are the most common errors which can be recognized in the output. Layout, formatting, and style errors are the subclasses of qualitative errors. These errors are latent errors from at least two perspectives. They can only be revealed when the file is opened, and their effects are recognizable when modification to the text is carried out.

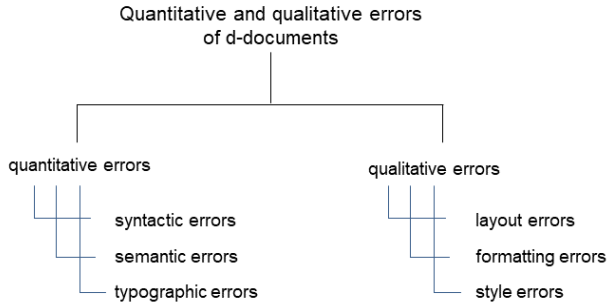


Figure 6: Classes and subclasses of quantitative and qualitative errors of d-documents.

In general, layout, formatting, and style errors originate from the d-characteristics and the design, syntactic and semantic errors from the language and content, and typographic errors from the printability of the texts.

Layout errors related to typing are relatively easily recognizable, in case, the creator is familiar with the definition of correctly formatted and edited text and the non-printable characters. Layout errors related to formatting, formatting errors in general, and distinguishing the origin of these errors require practice, consciousness, and accuracy, at the beginning of the learning process, teacher-guided activities.

Errors related to styles, in terms of their recognizability, are one of the most demanding qualitative errors. To decide and create styles for a document requires a clear concept of the problem, a well-built algorithm, a firm knowledge of the use of the software tools, and the ability to discuss and debug; in general, a deep approach problem solving method [39][17][15]. Beyond style design, styles require another skill, namely how to use them properly. Even if the styles are properly defined there is no guarantee that the user can apply them.

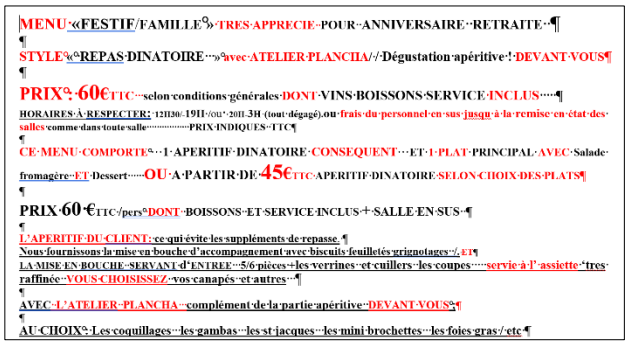
As mentioned in the introduction, typography is one further aspect of the language, since it can affect the readability of the text. Typographic errors can be categorized either as quantitative or qualitative errors, depending on the source and/or the result of these errors: most typographic tools are handled by formatting commands, however hard-coding would also lead to an incorrect printout.

Automated Error Recognition and Available Help

Beyond the above-mentioned tools for checking the correctness of the natural language of the documents (Section 2.3), end-users must also know that there is only limited automated help available which provides guidance. This deficiency can be explained, on the one hand, by the complexity of natural language texts, and on the other, by the surface approach methods which have been promoted by the big software companies, and unfortunately, unconditionally accepted by both end-users and teachers [15].

The surface approach methods usually are disguised by the 'user-friendly' slogan, which suggests that users, without any background knowledge, can create any kind of correct document in an extremely short period of time [28][20][17][32][15]. This problem is not unknown in pedagogy, in Chi et al. we can read (1982) that "Solving real-word problems presents new obstacles. … Basically, the exact operators to be used are usually not given, the goal state is sometimes not well defined, and more importantly, search in a large knowledge space becomes a serious problem.". With the 'all-mighty' digital tools and with the 'user-friendly' approaches both the knowledge space and the 'devoid space' are increased to a size for which our brain structure is not yet ready or never will be [32]. The problem is further deepened by help tools, which only focus on the handling of the interface by polishing the documents, without considering the structure of the texts [15]

In several cases the documents have the appearance of overelaborated structures, where there is no connection between the content and the final appearance [15]. In the teaching-learning process, it is a common practice to make students type meaningless texts and add unreasonable designs, a practice which does not develop the error-awareness of the end-users [15]. Apart from a very small number of weak attempts (the details of automated text correction are beyond the scope of the present paper), there has been no software-supported guidance and help for correction of qualitative errors.


MENU «FESTIF/FAMILLE» TRES APPRECIE POUR ANNIVERSAIRE RETRAITE

STYLE «REPAS DINATOIRE … » avec ATELIER PLANCHA / Dégustation apéritive ! DEVANT VOUS

PRIX : 60€ TTC selon conditions générales DONT VINS BOISSONS SERVICE INCLUS

HORAIRES À RESPECTER: 12H30/ 19H /ou/ 20H 3H (tout dégagé) ou frais du personnel en sus jusqu'à la remise en état des salles comme dans toute salle PRIX INDIQUES TTC

CE MENU COMPORTE … 1 APERITIF DINATOIRE CONSEQUENT … ET 1 PLAT PRINCIPAL AVEC Salade fromagère ET Dessert …… OU A PARTIR DE 45€ TTC APERITIF DINATOIRE SELON CHOIX DES PLATS

PRIX 60 € TTC /pers DONT BOISSONS ET SERVICE INCLUS + SALLE EN SUS

L'APERITIF DU CLIENT: ce qui évite les suppléments de repasse.

Nous fournissons la mise en bouche d'accompagnement avec biscuits feuilletés grignotages /. ET

LA MISE EN BOUCHE SERVANT d'ENTREE … 5/6 pièces + les verrines et cuillers les coupes …… servie à l'assiette "tres raffinée VOUS CHOISISSEZ vos canapés et autres …

AVEC L'ATELIER PLANCHA … complément de la partie apéritive DEVANT VOUS

AU CHOIX : Les coquillages … les gambas … les st jacques … les mini brochettes … les foies gras / etc

Figure 7. A sample with the appearance of overelaborated structures. These unnecessary and attention-distractor typographic solutions make the content difficult to comprehend. We also must note that the text is loaded with other quantitative and qualitative errors [48].

Especially when considering typography, with its two-fold features, there is no support at all to assist with the correctness of texts. Again, the users must be trained to handle their document in the correct way.

To make things even worse, the interfaces of the popular word processing programs offer typographically incorrect formatting options in a central position (for example: underline). Ignorant users and teachers focus on these formatting options just because of their central positions, without any thought for typography (Figure 7).

6. RESULTS

Based on the taxonomy used for classifying qualitative spreadsheet errors we have created the definition of the correctly edited text and a taxonomy of errors in end-user text management [H3]. We claim that errors can be classified into two major classes, quantitative and qualitative errors, and within each group further subgroups can be defined.

Quantitative errors are those which can be recognized in the printed form of the text. These are the hard-coded, categorized as syntactic, semantic, and typographic errors. Qualitative errors, which effect the operational use of d-documents are categorized as layout, formatting, and style errors.

Furthermore, we have given reasons and examples for the need for such a classification [H1]. The extremely high number of erroneous d-documents of the Internet and of private collections clearly prove that both the printout and operational use are violated by negligent text-handling. Without knowing what is correct and what is not in a d-document, bricolage is a self-serving use of editor programs instead of real text management.

We also claim that the presented error recognition model, instead of special and restricted interface-knowledge, highly supports knowledge transfer between the different areas of informatics, sciences and arts. Knowledge transfer based operations leads to effective concept based problem-solving, of which we are in great need.

Just as with the attempt to classify qualitative spreadsheet errors, we are aware that there is long way to reach a consensus, but our definition and classification would represent a starting point [H2].

7. CONCLUSIONS

We can conclude that before the actual word processing takes place, we must know what is correct and what is not. If we do not know the rules of end-user text management, we have no chance of doing this correctly. Beyond that, similar to problem solving in general [39] and especially to programming, the text management process must be planned in advance for the use of the software, and it must be followed with a discussion and debugging. Most importantly, the whole processes must take place before the dissemination of the document.

We also claim that education plays a crucial role in preparing students and end-users for conscious and effective text management. Those who teach the subject must be educated in informatics and computer sciences, sciences and arts, and pedagogy, to be able to prepare end-users for effective d-document handling. A lack of a firm theoretical background in any of these and their supplementary fields leads to teachers being unaware of their effects on students and, in general, to the creation and re-creation of erroneous documents and interface-centered teaching methods. In general, all the aspects of the TPCK (Technological Pedagogical Content Knowledge) [35] and the teachers' conscious belief in an "incremental" nature of science [6] must be present while teaching concept-based text management.

8. REFERENCES

[1] T. Bell & H. Newton, "Unplugging Computer Science", In **Improving Computer Science Education**. (Eds.) Djordje M. Kadijevich, Charoula Angeli, and Carsten Schulte. Routledge. 2013. pp. 66–81.

[2] M. Ben-Ari, "Bricolage Forever! PPIG 1999", 11th Annual Workshop. 5–7 January 1999. **Computer-Based Learning Unit**, University of Leeds, UK. Retrieved: 12. 03. 2015. from http://www.ppig.org/papers/11th-benari.pdf.

[3] M. Ben-Ari & T. Yeshno, „Conceptual models of software artifacts", **Interacting with Computers**, Volume 18, Issue 6, 2006, pp. 1336–1350.

[4] P. L. Bewig, "How do you know your spreadsheet is right? Principles, Techniques and Practice of Spreadsheet Style", 2005. Retrieved: 12. 11. 2018. from https://arxiv.org/ftp/arxiv/papers/1301/1301.5878.pdf.

[5] Gy. Bujdosó.& M. Csernoch, "Digital Literacy, Digital Language", In Hungarian: "Digitális írástudás, digitális nyelvhelyesség", Tudományos és Műszaki Tájékoztatás 61:(10), 1–10. 2014.

[6] J. A. Chen, D. B. Morris, & N. Mansour, "Science Teachers' Beliefs. Perceptions of Efficacy and the Nature of Scientific Knowledge and Knowing", **In International Hand-book of Research on Teachers' Beliefs,** (Eds.) Fives, H. & Gill, M. G. Routledge, 2015, pp. 370–386.

[7] M. Chi, R. Glaser, & E. Rees, "Expertise in problem solving", In Sternberg, R. (ed.), **Advances in the Psychology of Human Intelligence**, Erlbaum, Hillsdale, NJ, 1982, pp. 7–75.

[8] D. C. Francis, L. Rapacki, & A. Eker, "The Individual, the Context, and Practice", **In International Handbook of Research on Teachers' Beliefs**, (Eds.) Fives, H. & Gill, M. G. Routledge, 2015, pp. 336–352.

[9] M. Csernoch, "Methodological Questions of Teaching Word Processing", **3rd International Conference on Applied Informatics**, Eger-Noszvaj, Hungary, 1997, pp. 375–382.

[10] M. Csernoch, "Newly introduced word-types and lemmas in Dan Brown's The Da Vinci Code and its translations", **Across Languages and Cultures**, Jun 2007, Vol. 8, Issue 2. pp. 195–220.

[11] M. Csernoch, "Teaching word processing – the theory behind", **Teaching Mathematics and Computer Science**. 2009/1, pp. 119–137.

[12] M. Csernoch, "Teaching word processing – the practice", **Teaching Mathematics and Computer Science**, 2010, (8) 2, pp. 247–262.

[13] M. Csernoch, "Clearing Up Misconceptions About Teaching Text Editing", In: I Candel Torres L Gómez Chova A López Martínez (ed.) ICERI2011: **4th International Conference of Education, Research and Innovation**. Madrid, Spain, (IATED), 2011, pp. 407–415.

[14] M. Csernoch, **Programming with Spreadsheet Functions**, In Hungarian: **Programozás táblázatkezelő függvényekkel – Sprego**, Műszaki Könyvkiadó, Budapest, 2014.

[15] M. Csernoch, "Thinking Fast and Slow in Computer Problem Solving", **Journal of Software Engineering and Applications**, Vol.10 No. 01(2017), Article ID:73749, 30 pages 10.4236/jsea.2017.101002.

[16] M. Csernoch, & P. Biró, "Sprego programming", **Spreadsheets in Education (eJSiE)** (8) 1. Retrieved 12/04/2015 from http://epublications.bond.edu.au/cgi/viewcontent.cgi?article=1175&context=ejsie, 2015.

[17] M. Csernoch, & P. Biró, "Computer Problem Solving", In Hungarian: "Számítógépes problémamegoldás", **TMT, Tudományos és Műszaki Tájékoztatás**, Könyvtár- és információtudományi szakfolyóirat, (62) 3, 2015, pp. 86–94.

[18] M. Csernoch, & P. Biró, "Wasting Human and Computer Resources", World Academy of Science, Engineering and Technology, International Science Index 98, **International Journal of Social, Education, Economics and Management Engineering**, 2015, 9(2), pp. 564–572.

[19] M. Csernoch, & E. Dani, "Data-structure validator: an application of the HY-DE model", In: **IEEE 8th International Conference on Cognitive InfoCommunications: CogInfoCom**. Debrecen, Hungary, 2017.09.11-2017.09.14. (IEEE) Piscataway (NJ): IEEE Computer Society, 2017. pp. 197–202. (International Conference on Cognitive Infocommunications) (ISBN:978-1-5386-1264-4).

[20] ECDL 2014. "The Fallacy of the 'Digital Native': Why Young People Need to Develop their Digital Skills", Retrieved 18/12/2017 from http://ecdl.org/media/thefallacy-ofthe'digitalnative'positionpaper1_1.pdf.

[21] EuSpRIG Horror Stories. Retrieved 18/07/2017 from http://www.eusprig.org/horror-stories.htm

[22] K. Freiermuth, J. Hromkovic, & B.Steffen, "Creating and Testing Textbooks for Secondary Schools. In Informatics Education - Supporting Computational Thinking", Third International Conference on Informatics in Secondary Schools - Evolution and Perspectives, **ISSEP 2008** Torun Poland, July 1-4, 2008 Proceedings. Mittermeir, Roland, Syslo, Maciej M. (Eds.)

[23] J. Gould, "Some psychological evidence on how people debug computer programs" **International Journal of Man-Machine Studies**, 1975, (7) 1, pp. 151–182.

[24] J. Gould, & P. Drongowski, An exploratory study of computer program debugging. Human Factors, 1974, 16, pp. 258–277.

[25] M. Gove, Michael Gove speech at the BETT Show 2012. Retrieved: 12. 03. 2015. from https://www.gov.uk/government/speeches/michael-gove-speech-at-the-bett-show-2012.

[26] O. Hazzan, T. Lapidot, & N. Ragonis, **Guide to Teaching Computer Science: An Activity-Based Approach**, Springer, 2011.

[27] J. Hromkovic, **Algorithmic Adventures – From Knowledge to Magic**, Springer, 2009.

[28] IEEE&ACM Report 2013. Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. December 20, 2013. The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society. Retrieved 12/03/2015 from http://www.acm.org/education/CS2013-final-report.pdf.

[29] L. Jerinic, "Teaching Introductory Programming Agent-based Approach with Pedagogical Patterns for Learning by Mistake", (IJACSA) **International Journal of Advanced Computer Science and Applications**, 2014, (5) 6.

[30] D. Kahneman, **Thinking, Fast and Slow**, New York: Farrar, Straus; Giroux, 2011.

[31] P.A. Kirschner, J. Sweller, & R.E. Clark, "Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist Discovery", **Problem-Based, Experiential, and Inquiry-Based Teaching,** Educational Psychologist, 2006, 41(2), pp. 75–86.

[32] P. A. Kirschner, & P. De Bruyckere, "The myths of the digital native and the multitasker", **Teaching and Teacher Education**, (2017), 67 pp. 135–142.

[33] J. Kruger, & D. Dunning, "Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments", **Journal of Personality and Social Psychology,** 1999, 77 (6), pp. 1121–34.

[34] L. Leon, Z. Przasnyski, & K. C. Seal, "Introducing a Taxonomy for Classifying Qualitative Spreadsheet Errors", **Journal of Organizational and End User Computing**, 2015, 27 (1), pp. 33–56.

[35] P. Mishra & M. J. Koehler, "Technological Pedagogical Content Knowledge: A new framework for teacher knowledge". **Teacher College Record**, 2006, 108(6), pp. 1017–1054.

[36] R. R. Panko, "What We Know About Spreadsheet Errors" **Journal of End User Computing**. Special issue on Scaling Up End User Development, 2008, (10) 2, pp. 15–21.

[37] R. R. Panko "The Cognitive Science of Spreadsheet Errors: Why Thinking is Bad", **Proceedings of the 46th Hawaii International Conference on System Sciences**, January 7-10, 2013, Maui, Hawaii.

[38] R. R. Panko & S. Aurigemma, "Revising the Panko-Halverson taxonomy of spreadsheet errors", **Decision Support Systems**, 2010, (49) 2, pp.235–244.

[39] Gy. Pólya, **How To Solve It,** A New Aspect of Mathematical Method. Second edition. Princeton University Press, Princeton, New Jersey, 1957.

[40] Z. Przasnyski, L. Leon, & K. C. Seal, "In Search of a Taxonomy for Classifying Qualitative Spreadsheet Errors", **Proceedings of EuSpRIG 2011 Conference**, "Spreadsheet Governance – Policy and Practice", 2011.

[41] J. Reason, **Human Error**, Cambridge University Press, 1990.

[42] G. Reynolds, **Presentation Zen: Simple Ideas on Presentation Design and Delivery**, (2nd Edition) (Voices That Matter). New Riders, 2011.

[43] The Unicode Consortium. Retrieved 27/12/2017 from http://unicode.org/.

[44] Unicode 10.0 Character Code Charts. Retrieved 27/12/2017 from http://www.unicode.org/charts/. http://www.unicode.org/charts/charindex.html.

[45] C0 Controls and Basic Latin. Retrieved 27/12/2017 from https://www.unicode.org/charts/PDF/U0000.pdf.

[46] Unicode Blocks. Retrieved 27/12/2017 from http://www.fileformat.info/info/unicode/block/index.htm.

[47] J. M. Wing, "Computational Thinking", **Communications of the ACM**, 2006, (49) 3.

9. SOURCES

[48] Author: N/A (2016) MENU «FESTIF/FAMILLE » Retrieved 27/12/2017 from http://www.lamusardiere-foucart.fr/modules/uploadmanager11/admin/index.php?action=file_download&file_id=94.

[49] P. Sammons, "The Impact of Pre-School on Young Children's Cognitive Attainments at Entry to Reception", Retrieved 27/11/2018 from Sylvadiscovery.ucl.ac.uk/10005280/1/FINAL_BERJ_ARTICLE_10_March_04.doc.

[50] D. Henderson, **Parent Handbook. Welcome to Allendale Pre-school Playgroup….**, Retrieved 27/11/2018 from https://allendalepreschool.files.wordpress.com/2014/01/parent-handbook1.docx.