

MDIR Constructor: Interactive digital activities for developing reading and writing skills with natural numbers

Nicolae BALMUȘ, Tatiana CHIRIAC

“Ion Creangă” State Pedagogical University of Chișinău,
Chisinau, Republic of Moldova

balmus.nicolae@upsc.md, chiriac.tatiana@upsc.md

Abstract: *The article presents MDIR Constructor, a digital authoring environment for creating interactive digital textbooks and learning activities across different school subjects, with a particular focus on mathematics. The system includes modules such as NumeralsMathLab and NumeralsVirtLab, which help students to develop reading, writing, and numerical literacy by converting numbers to words, self-assessment tasks, and mathematical dictations that provide instant feedback. The incorporation of Python libraries and AI tools into the Delphi programming environment enhances system adaptability, scalability, and multilingual support. By combining interactive digital resources with traditional curricular objectives, MDIR Constructor provides meaningful learning experiences, supports real-time feedback, and improves students' fluency in numerical literacy. The paper also discusses the implications of integrating artificial intelligence and interactive digital tools into mathematics education, highlighting opportunities for adaptive and personalized learning.*

Keywords: Educational software, Interactive digital learning, Numerical literacy, Self-assessment, Mathematical dictation, Python libraries, AI tools in education.

1. Introduction

MDIR Constructor is software that allows users to build and develop interactive digital textbooks with editable content for any school subject. The copyright and related rights for MDIR Constructor 3.0 have been registered with AGEPI Republic of Moldova (Balmuș & Chiriac, 2024). MDIR Constructor has recently been enhanced with interactive digital challenges tailored to mathematics school textbooks. The present research describes several interactive digital activities designed to accomplish learning/assessment tasks outlined in fifth-grade mathematics school textbooks from the Republic of Moldova (Achiri et al., 2020) and Romania (Voiculescu et al., 2022). The programming code of the applications is written in the Delphi 12 (2025) programming environment with the free Python4Delphi (2025) component set and some TMS Software package components included (TMS Software, 2025).

The impact of using Python libraries in the Delphi programming environment is significant and is manifested by the significant expansion of the functionalities of both languages, based on which programmers create interactive

<https://doi.org/10.58503/icvl-v21y202629>

applications with a relevant graphical interface (Delphi's strength) through concise and high-performance programming code (Python's strength). Access to Python libraries in MDIR Constructor environment is achieved through a Python code interpreter with a visual interface, the programming code of which we present below.

2. Implementing a Python code interpreter in Delphi applications

To use Python subroutine libraries within the Delphi programming environment, a recent version of the Python language must first be installed on the user's computer (in this case, Python 3.13.9-amd64 (Python 3.13, 2025) was used). The structure and interface of the *MyPython* application, integrated into MDIR Constructor, are shown in Figure 1a, while the output resulting from the execution of the Python code is presented in Figure 1b.

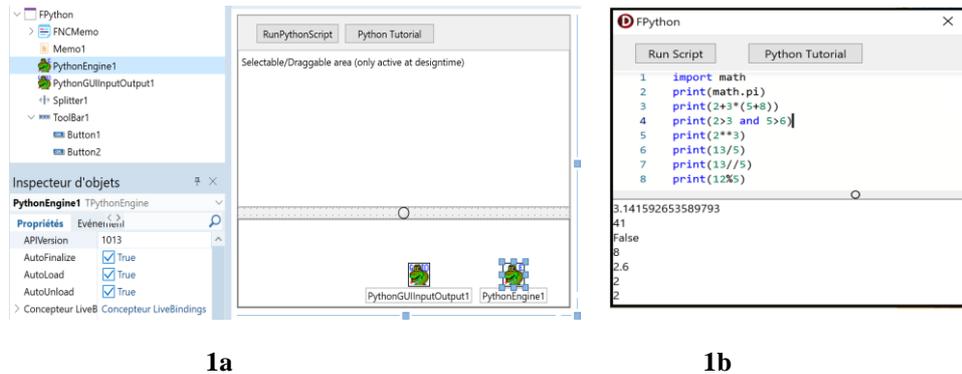


Figure 1. Application design (a) and the result of executing the Python code (b)

The main code of the application is implemented in the *OnClick* event handler of the *Button1* component:

```
procedure TFPython.Button1Click(Sender: TObject);
begin
  memo1.Lines.Clear;
  PythonEngine1.ExecStrings(FNCMemo.Lines);
end;
```

When executed, the user writes the Python code in the *FNCMemo* component window and presses the *RunPythonScript* button. If the code is correct, the result is displayed in the *Memo1* window.

If the programming code contains errors, the *Memo1* window also displays the errors and correction suggestions (see Figure 2).

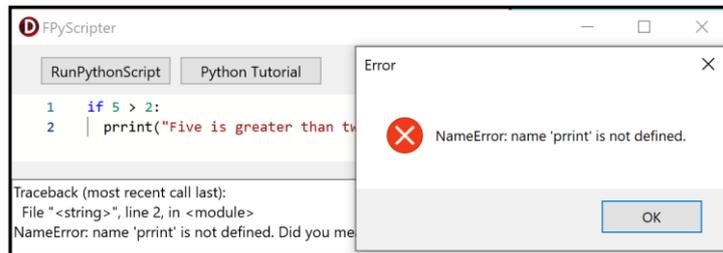


Figure 2. Managing Python programming errors

To ensure proper syntax handling and visualization of the Python language, the *Language* property of the *FNCMemo:TTMSFNCMemo* component was set to the value *mIPython*. As a result, Python keywords and code blocks are correctly highlighted (see Figure 3).

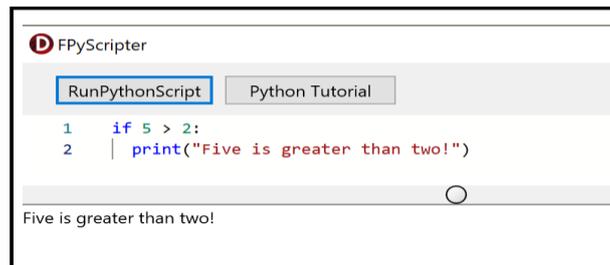


Figure 3. Example of Python code block highlighting

In the MDIR Constructor software, the *PyScripter* module is used in two ways:

1. As a standalone application for carrying out algorithmic and programming activities described in school informatics textbooks;
2. As a module for performing interactive calculations for various functions implemented in other modules.

Below is an example illustrating the use of Python libraries within the Delphi 12 programming environment.

3. Converting numbers into word strings

The conversion of numbers (integers or decimals) into word strings and vice versa is a common activity presented in school textbooks for mathematics as well as for native or foreign language studies. Through these activities, students develop reading and writing skills related to numbers.

To perform such activities in an interactive digital format within educational software, subroutines are required to convert a numerical value into a string in strict accordance with the orthography of the intended natural language. These subroutines must also be compatible with the programming language in which the educational software is developed.

The MDIR Constructor software is developed in the Delphi programming environment. For converting natural numbers into word strings in accordance with Romanian orthography, a subroutine written in the Pascal language was implemented, which operates correctly for numbers ranging from 0 to 999 billion. This subroutine consists of approximately 450 lines of code.

However, adapting it for number-to-word conversion in another programming language requires a considerable amount of time. This problem can be solved much more efficiently by using Python libraries. One of the most popular libraries for this purpose is *num2words*, which can be installed using the command: `pip install num2words`.

The access to the library's functions is achieved through the following instruction: `from num2words import num2words`.

The *num2words* function can be called with one or more parameters (e.g. Figure 4).

The screenshot shows the FPyScripter interface. On the left, there is a code editor with the following Python code:

```

1 from num2words import num2words
2 # Most common usage.
3 print(num2words(36))
4 print(num2words(36.5, lang='ro'))
5 # Other variants, according to the type of article.
6 print(num2words(36, lang='ro', to='ordinal'))
7 print(num2words(36, lang='ro', to='ordinal_num'))
8 print(num2words(36, lang='ro', to='year'))
9 print(num2words(36, lang='ro', to='currency'))

```

On the right, there is a text area displaying the output of the code, with an arrow pointing from the code to the output:

```

thirty-six
treizeci și șase virgulă cinci
al treizeci și șaselea
al 36-lea
treizeci și șase
treizeci și șase de lei

```

Figure 4. Structure of the *num2words* function parameters

The first parameter (the numerical value) is mandatory. In this case, the result is displayed in English. If the *lang* parameter is specified, the result is displayed in the selected language. Based on these investigations, the *PyNaToWords* function was implemented in Pascal code and can be applied across all modules of the MDIR Constructor software.

```

Function PyNaToWords(sender:TObject;N:int64;lg:string):string;
var s,ns:string;
begin
  FPyScripter.FNCMemo.Lines.Clear; FPyScripter.memo1.Lines.Clear;
  FPyScripter.FNCMemo.Lines.Append('import math');
  FPyScripter.FNCMemo.Lines.Append('from num2words import
num2words');
  ns:=n.ToString;
  s:='num2words('+n.ToString+', lang='+lg+'';
  FPyScripter.FNCMemo.Lines.Append('print('+s+')');
  FPyScripter.Button1.OnClick(sender);
  result:=FPyScripter.Memo1.Text;
end;

```

The following section presents the *NumeralsMathLab* module, implemented in the MDIR Constructor software, which allows users to carry out activities aimed at strengthening their knowledge of reading, writing, and converting natural numbers.

4. NumeralsMathLab – creation and application activities

NumeralsMathLab is a submodule of the *Math5aLab* module that allows users to do interactive digital activities that correspond to the content of fifth-grade mathematics textbooks. The fifth-grade mathematics textbooks in the Republic of Moldova and Romania begin with the topic *§1 Reading and Writing Natural Numbers*, where the rules for reading and writing natural numbers are presented in tabular style, as seen in Figure 5.

CLASA MILIARDELOR			CLASA MILIOANELOR			CLASA MILOILOR			CLASA UNITĂȚILOR			Cum citim?								
S	Z	U	S	Z	U	S	Z	U	S	Z	U									
									7	0	3	șapte sute trei								
								1 3	0	8	5	treisprezece mii optzeci și cinci								
									5	9	1	6	cinci sute nouăzeci și unu de milioane șapte mii patru sute șaisprezece							
									6	0	4	5	2	0	7	6	0	0	8	șaiszeci de miliarde patru sute cincizeci și două de milioane șaptezeci și șase de mii opt

CLASA MILIARDELOR			CLASA MILIOANELOR			CLASA MILOILOR			CLASA UNITĂȚILOR				
sute de miliarde	zeci de miliarde	unități de miliarde	sute de milioane	zeci de milioane	unități de milioane	sute de mii	zeci de mii	unități de mii	sute	zeci	unități		
			3	1	8	5	2	0	2				
			trei milioane	o sută optzeci și cinci de mii			doi sute doi						
NUMĂR NATURAL													

Figure 5. Rules for Writing Natural Numbers

According to Figure 5, students should learn to read and write in Romanian for natural numbers ranging from 0 to 999 billion. Two or three practice activities are supplied. However, it is clear that this information alone is insufficient for acquiring fluent reading and writing skills with natural numbers. The Romanian spelling for writing numerals is fairly difficult, and it is still studied in Romanian language classrooms from grades V to VII (Curriculum Național, aria curriculară limbă și comunicare. Disciplina limba și literatura română. Clasele V – IX, 2019). The study of numerals is also included in the foreign language curriculum (Curriculum Național, aria curriculară limba străină. Clasele V – IX, 2020).

Given this context, the MDIR Constructor software has been enhanced with the *NumeralsVirtLab* module, which allows users to improve their numerical writing and reading abilities. In addition to the functions described in the previous section, the development of this module also integrates artificial intelligence technologies that have been integrated into the components of the TMS Software package (TMS Software, 2025).

The interface of the application in which the user performs learning activities related to reading and writing numbers in words is shown in Figure 6.

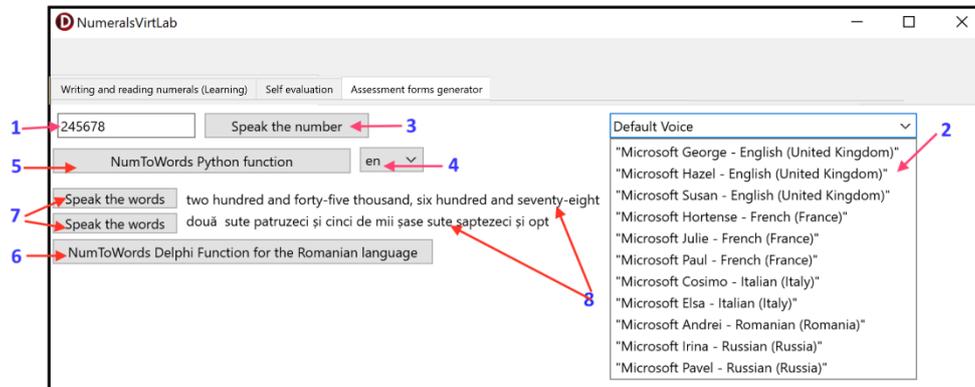


Figure 6. Reading and Writing Natural Numbers

To carry out the learning activities, the user follows these steps:

1. In the editing field (1), enter the number (from 0 to 999 billion);
2. In the *ComboBox* (2), select the language in which the number will be pronounced;
3. Press button (3) to listen to the number pronounced by a voice generated using AI tools;
4. Select one of the languages (en – English, fr – French, ro – Romanian, ru – Russian) to be used for displaying the number written in words;
5. Press button (5);
6. In field (7), the number written in words appears in accordance with the language selected in step 4, generated by the *NumToWordsPython* function;
7. Press button (6). As a result, the number written in words in Romanian appears, generated by the *NumToWordsDelphi* function;
8. If the user wants to listen to the pronunciation of the numbers generated by these functions, buttons (7) and (8) can be used.

The Romanian language has a complex orthography for writing numbers in words, and the programming functions developed for this activity may, in some cases, produce errors. An example of such an error is shown in Figure 7.

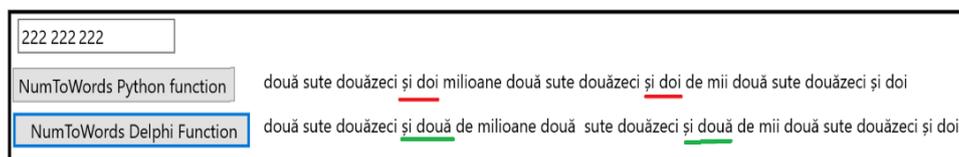
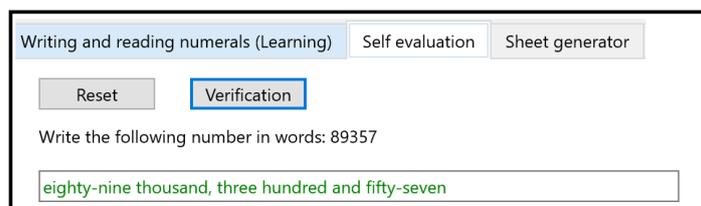


Figure 7. Example of a Conversion Error

On the *Self-Assessment* page, the user performs self-evaluation activities:

Activity 1. Writing Natural Numbers in Words

The interface of the application for this activity is shown in Figure 8.



Writing and reading numerals (Learning) Self evaluation Sheet generator

Reset Verification

Write the following number in words: 89357

eighty-nine thousand, three hundred and fifty-seven

Figure 8. Example of a Self-Assessment activity

When the *Reset* button is pressed, a number appears that must be written in words. In the text field designated for editing the answer, the user types the corresponding word string using the keyboard, representing the conversion of the number into words. If the task is completed correctly, pressing the *Check* button displays the entered text in green. In case of an incorrect answer, the text is displayed in red.

Activity 2. Writing Natural Numbers with Digits

The interface of the application for this activity is shown in Figure 9. When the *Reset* button is pressed, a task appears that requires converting a number from a word string format into numeric form. In the designated field, the user enters the number using digits. If the task is completed correctly, pressing the *Check* button displays the entered number in green. In case of an incorrect answer, the text is displayed in red. An example is presented in Figure 9.



Reset Verification

Write the following number with digits:

fifty-four thousand, eight hundred and forty-seven

54847

Figure 9. Example of a Self-Assessment activity

Activity 3. Mathematical Dictations

Mathematical dictation is a popular approach to assessing student knowledge in primary and lower secondary schools. In typical situations, students listen to the task, write their responses on paper, and then the teacher analyses them. The results of these assessments typically get delivered to students after a particular period of time.

Students can use the *NumeralsMathLab* application to do mathematical dictation-type activities and receive immediate feedback. Figure 10 illustrates an example of such an activity. When the *Reset* button is pressed, an assignment created with AI technologies is delivered verbally. The *Repeat* button allows the

task to be replayed to enhance understanding. The student afterwards writes the result into the designated space. If the response is true, the feedback is immediate: the entered text is displayed in green for a correct result and red for a wrong one.

Figure 10. Example of a “Mathematical Dictation” activity

Currently, the application generates tasks of the following types:

- Write in digits the number that is heard;
- Find the sum, difference, or product of two natural numbers;
- Find the quotient and remainder of the division of two natural numbers;
- Determine the number that is greater by n_1 than twice/triple the number n_2 ;
- Etc.

In general, various types of tasks can be implemented, as the system generates numerical data randomly and integrates it into dynamically created problem statements. The created text is read aloud using the speech synthesizer integrated into MDIR Constructor.

The *Assessment Form Generator* page is only accessible to administrators. This page allows the teacher to generate randomized evaluation sheets that can be printed on paper. The application allows the user to specify the number of variants and elements included in each assessment. Figure 11 displays an example of an evaluation sheet.

Figure 11. Example of an evaluation sheet with randomized content

5. Conclusions

The MDIR Constructor software offers a versatile platform for designing interactive digital learning activities that help students strengthen their reading, writing, and math skills. The software integrates modules such as *NumeralsMathLab* and *NumeralsVirtLab*, allowing students to practice converting numbers between digits and words, do self-assessment exercises, and participate in mathematical dictation activities with immediate feedback. The use of Python libraries and AI-powered tools enhances the flexibility and scalability of these activities, making it possible to adapt them to multiple languages and diverse educational contexts. Overall, integrating interactive digital tools with traditional learning objectives fosters more engaging and effective learning experiences, provides immediate feedback, and supports the gradual development of fluency in numerical literacy. Future developments could expand the system's coverage of mathematical topics and incorporate additional AI-driven adaptive learning methodologies, further enhancing personalized learning opportunities for students.

6. Limitations and future work

It should be noted that the present research represents an initial stage of an ongoing development and research project. At this stage, the contribution primarily focuses on the design and technical implementation of interactive digital mathematics activities within the MDIR Constructor environment.

In relation to existing solutions for interactive digital textbooks, MDIR Constructor distinguishes itself by enabling the integration of Python-based interactive components within a Delphi graphical interface, allowing for editable, curriculum-aligned content creation without requiring advanced programming expertise from end users.

The educational impact evaluation, including systematic user feedback and classroom-based implementation scenarios, is planned as a subsequent phase of the research. Future work will include pilot testing in real classroom settings, collection of teacher and student feedback, and analysis of learning outcomes in order to empirically validate the pedagogical effectiveness of the proposed solution.

Acknowledgments

This study was conducted within the framework of the research project “Innovation in the Design and Implementation of Interactive Digital Math Textbooks for Lower Secondary Education”, Strategic Direction: “Innovative Educational Technologies and Products”, registered in the State Register of Projects in the Field of Science and Innovation of the Republic of Moldova, project code 25.80012.0807.42SE.

REFERENCE

- Achiri, I., Braicov, A., Șpunteco, O., Ursu, L. (2020) *Matematică: Manual pentru clasa a 5-a*. Ministerul Educației, Culturii și Cercetării al Republicii Moldova. – Chișinău: Prut Internațional. 228 p.
- Balmuș, N. & Chiriac, T. (2024) MDIR Constructor 3.0 – software pentru crearea manualelor digitale interactive cu content redactabil. Agenția de Stat pentru Proprietatea Intelectuală (AGEPI) a Republicii Moldova. Seria: PC (program pentru calculator). <https://www.db.agepi.md/opere/Details.aspx?RealID=7285&lang=ro> [Accessed: 28 October 2025].
- Curriculum Național. (2019) *Aria curriculară limbă și comunicare. Disciplina limba și literatura română. Clasele V – IX*. Ministerul Educației, Culturii și Cercetării al Republicii Moldova. https://mecc.gov.md/sites/default/files/limba_si_lit_romana_curriculum_sc_nat_gimnaziu.pdf [Accessed: 26 October 2025].
- Curriculum Național. (2020) *Aria curriculară limba străină. Clasele V – IX*. (2020) Ministerul Educației, Culturii și Cercetării al Republicii Moldova. https://mecc.gov.md/sites/default/files/limba_straine_gimnaziu.pdf [Accessed: 26 October 2025].
- Embarcadero Delphi12 (2025) <https://www.embarcadero.com/products/delphi/starter> [Accessed: 20 September 2025].
- Embarcadero Python4Delphi. (2025) IDE: Delphi 13. Edition: See IDE. <https://getitnow.embarcadero.com/python4delphi/> [Accessed: 20 September 2025].
- MathTools. (2025) Number to words Converter. Available at: <https://math.tools/calculator/numbers/words> [Accessed: 25 October 2025].
- Python313. (2025) <https://www.python.org/downloads/release/python-3130/2025> [Accessed: 20 September 2025].
- TMS Software. (2025) <https://www.tmssoftware.com/site/tmsallaccess.asp> [Accessed: 20th September 2025].
- Voiculescu, D.-I., Dumitrescu, A.-I., Gheorghe, A., Popa, C.-S., Stancu, F. & Voiculescu, C.-V. (2022) *Matematică: manual pentru clasa a V-a*. Ministerul Educației al României. Editura - Pitești: Paralela 45.