AUTOMATION OF CADASTRE WORKS USING THE LISP PROGRAMMING LANGUAGE

A. ȘMULEAC[1], A. BUDA[1], M. HERBEI[1], Laura ȘMULEAC[1], G. POPESCU[1], R. PAȘCALĂU[1]

[1] University of Life Sciences "King Mihai I" from Timisoara, 300645, 119, Calea Aradului, Timisoara, Romania

Corresponding author: laurasmuleac@usvt.ro

Abstract. *Time is a resource never enough or at hand in the most important moments, a resource that we are forced to rationalize. No matter how hard we try, the time itself will not be manageable, only the activities that are carried out in a certain period. Planning and staging the way of working is a method of ensuring control for the fulfilment of all tasks necessary to complete the general cadastre works, but also removing distractions from the objective, thus obtaining optimization at management level. The use of AutoCAD software for cadastre works has become more and more frequent with the passage of time, thus facilitating the development of new applications that can be loaded and used within the basic software. These applications are written using the AutoLISP programming language developed by Autodesk, which is derived from the LISP programming language. Knowing how to work to complete the general cadastre works, we can deduce the stages that can be automated within the AutoCAD software and develop applications according to the needs of each stage. This paper aims to use Visual Studio Code software to write and edit application code customized according to identified needs, but also by testing applications within AutoCAD software and presents how to work within a general cadastre project, carefully following each stage in order to automate them.*

Keywords*: time, AutoCAD, AutoLISP, Visual Studio Code, CGXML, general cadastre*

INTRODUCTION

For today's society, time has become one of the most important resources. Time costs, it is a value in itself. No one can afford to waste time (CUCOS, 2023).

Superficial management or lack of management turns time, or rather, lack thereof, into the main stress factor and cause of various ailments, which lead to aging of the body long before its time. For this reason, time management highlights a vital issue both professionally and personally, constantly influencing quality of life (https://thewoman.ro).

In this context, planning and staging the way of working is a method of optimizing time to accomplish all the tasks necessary to complete the general cadastre works, ultimately leading to time optimization and good quality management (https://learn.microsoft.com/ro).

The existence of the general cadastre on administrative territories and the creation of related cadastral databases, as well as their assembly into cadastral databases at the level of counties and the entire country, requires time, funding, technical resources and cadastre specialists (OPREA ET ALL, 2013).

Given that real estate transactions have intensified with the passage of time and land restitution, this has imposed a fairer record of real estate in terms of cadastre. The importance of the cadastre lies in the fact that through it are provided real information regarding real estate - position, size, use, owner - information necessary in all branches of the national economy.

The land register was first introduced in Austria and then in other European countries. Currently, this advertising system applies in Austria, Germany, Switzerland, Czech Republic, Slovakia, Poland, Romania, Italy, Germany. On the territory of our country they were introduced in Transylvania and Bukovina, gradually (FAINITA, 2016).

After 1930, the Bonne cartographic projection system was adopted both for the cadastre and for the needs of the army, stereographically 1930 with a point in Brasov. By Law nr. 23/1933, then amended by Decree-Law no. 115/1938 for the unification of the provisions regarding land books, the organization of the general cadastre and modern real estate advertising in Romania was established, starting from unitary geodetic networks, with cadastral plans and registers elaborated in a unitary system throughout Romania. At first, cadastral works were planned in Muntenia and Dobrogea, then in Muntenia and Oltenia, as well as updating the existing ones in Transylvania, Banat and northern Bukovina, along with them it was foreseen the introduction of the land books system (DECREE-LAW No. 115 of April 27, 1938).

The role of cadastre in economic development and the establishment of relations specific to a modern rule of law state is now officially recognized in our country. The two public institutions, subordinated to different ministries, have clearly defined fields of activity and functions that intertwine and condition each other. The normal functioning of this ensemble is constituted as an instrument for defending and guaranteeing private property, the basis for the development of society in a modern democracy and becomes an obligation in the European community (BADEA, 2013).

The basic problem of the general cadastre of obtaining plans must be seen from the perspective of general interests, with sustainable solutions of perspective, which ultimately lead to favorable economic effects.

The cartographic basis necessary for the introduction of the general cadastre must be ensured as a priority, in a short time, with minimal expenses and in compliance with qualitative conditions, the resolution of which raises serious technical and economic difficulties.

Photogrammetry allows well-defined technological processes to obtain topo-photogrammetric plans with higher efficiency, in a shorter time, but with less precision, which include many of the elements necessary for the technical, quantitative cadastre (SABĂU, CRAINIC, 2006).

In carrying out the technical function of the cadastre, besides the works specific to geodesy, topography and photogrammetry, there are technical problems specific to the cadastre, such as: detachment of surfaces, calculation of surfaces, rectification of borders, intersections of lines, parallel and perpendicular lines, segment points (BOȘ, IACOBESCU, 2009).

MATERIAL AND METHODS

For carrying out general cadastre works, time is an essential factor where the approach to the way of working makes the difference in terms of its completion on time. That is why automation is an essential component in the execution process of cadastre works, implemented at each stage, we will know that the delivery deadline can be easier to meet due to reduced working times.

AutoCAD software also contains the AutoLISP programming tool, which is used to easily define different geometric shapes. AutoLISP uses default functions and codes to model parts of a drawing. The functions developed by the user also contain parts of the default functions and new commands written by users that help to define the new algorithm necessary to obtain the desired complex geometric shapes.

SOFTWARE APPLICATIONS USED:

1. LIPS, AutoLISP

LISP, an acronym for *LISt Processing*, is a programming language that has been designed for easy manipulation of data strings. Developed in 1959 by John McCarthy, it is a language commonly used for programming artificial intelligence (AI). It is one of the oldest programming languages still in relatively wide use (GREEN, 2008).

In LISP, all calculations are expressed as a function of at least one object. Objects can be other functions, data elements (such as constants or variables), or data structures. LISP's ability to calculate with symbolic expressions rather than numbers makes it convenient for artificial intelligence applications (GREEN, 2008).

AutoLISP is a programming language designed to extend and customize AutoCAD functionality. It is based on the LISP programming language, whose origins date back to the late 1950s.

AutoLISP was introduced as an application programming interface in AutoCAD version 2.1 in the mid-1980s. LISP was chosen as the programming interface for AutoCAD applications initially because it was uniquely suited to the unstructured design process of AutoCAD projects, which involved repeatedly trying different solutions to design problems.

The development of AutoLISP programs for AutoCAD is done by writing code in a text editor, then loading the code into AutoCAD and running it. Program debugging is managed by adding instructions to print the contents of variables at strategic points in the program. The developer of the application needs to figure out where in the program it is necessary to do this and what variables to look at. If it is discovered that there is not enough information to determine the error, it is necessary to go back and modify the code by adding more debug points. And, finally, when the program is made to work correctly, it will have to either comment or remove the debug code that was added later.

Figure 1 shows an application model written in the AutoLISP programming language and visualized using the Visual Code Studio text editor. This application converts MLINES objects into LWPolylines.

The advantage of using written functions in AutoLISP is that drawings will be obtained in a shorter time and in an easy way. These codes are saved in separate files and can be used whenever necessary.

```
(defun c:ml2pl ( / *error* doc el en i s1 s2 val var )
    (setq doc (vla-get-activedocument (vlax-get-acad-object))
          var '("CMDECHO" "PEDITACCEPT" "QAFLAGS")
          val (mapcar 'getvar var))
    )

    (if (setq s1 (ssget "_:L" '((0 . "MLINE"))))
        (progn
            (LM:startundo doc)
            (mapcar 'setvar var '(0 1 0))
            (repeat (setq i (sslength s1))
                (setq en (ssname s1 (setq i (1- i)))
                      el (entlast)
                      s2 (ssadd)
                )
                (command "_.explode" en)
                (while (setq el (entnext el))
                    (ssadd el s2)
                )
                (command "_.pedit" "_M" s2 "" "_J" "" "")
                (setq s2 nil)
            )
            (mapcar 'setvar var val)
            (LM:endundo doc)
        )
    )
    (princ)
```

Figure 1. Application code example written in AutoLISP programming language [source: Author]

Functions used in an AutoLISP routine are mathematical objects that return a numeric value from input data.

The AutoLISP routine is a simpler and concise programming language compared to the others and this makes it often used to create complex geometric shapes in AutoCAD. These functions contain different objects, such as: variables, constants, other functions, macros, classes, and object-oriented methods (AutoLISP Reference Guide, AutoCAD 2012 for MAC, 2011 Autodesk, Inc. All Rights Reserved).

An AutoLISP program consists of a series of expressions. AutoLISP expressions have the following

form:

(function

Arguments)

Each expression begins with an open bracket (left) and consists of a function name and optional arguments for that function. Each argument can also be an expression. The expression ends with a straight parenthesis. Each expression returns a value that can be used by a surrounding expression. The value of the last interpreted expression is returned to the call expression. [1]

For example, the following sample code involves three functions:

(FUN1 (FUN2

Arguments)(FUN3

(arguments)

If you enter this code into the AutoCAD command line, the AutoCAD AutoLISP interpreter processes the code. The first function, fun1, has two arguments, and the other functions, fun2 and fun3, each have an argument. The functions fun2 and fun3 are surrounded by the fun1 function, so their return values are passed to fun1 as arguments. The fun1 function evaluates the two arguments and returns the value in the window from which you entered the code.

Knowing the above-mentioned notions, it will be necessary to analyze the way of work performed for the execution of general cadastre works, so that later the stages where automation can be implemented using functions from AutoLISP and creating functions will be identified.

2. AutoCAD

AutoCAD is a computer-aided design (CAD) software developed by Autodesk Inc. California, USA. This software is used by designers from several fields, such as: mechanics, robotics, electronics, electromechanics, civil engineering, architecture, industrial design, etc. With a user-friendly interface rich in drawing and modeling tools, AutoCAD offers the possibility to realize detailed projects, both in 2D and 3D (https://www.itlearning.ro).

The AutoCAD platform underpins the development of specialized Autodesk programs for advanced design in various fields:

- mechanical (AutoCad Mechanical)
- electric (AutoCAD Electrical)
- architecture (AutoCAD Architecture)
- civil engineering (AutoCAD Civil 3D)
- cartography (AutoCAD Map 3D)
- factory design (AutoCAD Plant 3D, AutoCAD P&ID).

The facilities related to the implementation of the geographical location (Geographic Location) allow the extension of the design activity in the field of cartography and spatial planning, making direct connection with real geospatial coordinates and satellite images (https://www.itlearning.ro).

3. Visual Studio Code

Visual Studio Code (https://code.visualstudio.com/) is a lightweight yet powerful source code editor that runs on the desktop and is available for Windows, MacOS, and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C+, C#, Java, Python, PHP and Go as well as runtimes (such as .NET and Unity). It allows you to expand your capacity through extensions. Visual Studio Code extensions can add more features to the overall experience, making it easier for programmers (https://docs.microsoft.com/).

The autocomplete capability in the Visual Studio Code extension shows the current context being edited and the relevant AutoComplete items through IntelliSense.

IntelliSense (https://code.visualstudio.com/docs/editor/intellisense) is a general term for various code editing functions, including: code completion, parameter information, quick information, and member lists. IntelliSense features are sometimes called by other names, such as "code completion," "content assist," and "code indication."

IntelliSense features (ROUSE, 2015) are powered by a language service that provides intelligent code completions based on language semantics and source code analysis. If a language service is aware of possible additions, suggestions will appear as you type. If you continue to enter characters, the list of members (variables, methods) is filtered to include only members that contain the entered characters. Pressing *Tab* or *Enter* will insert the selected member.

4. Microsoft Office Excel

Excel is a very powerful tool to extract meaning from large volumes of data. It also works very well for simple calculations and to keep track of almost any type of information. The key to harnessing your full potential is the grid of cells. Cells can contain numbers, text, or formulas. We can put the data into cells and group it into rows and columns. This will allow adding data, sorting and filtering, putting in tables and building great-looking charts.

Excel documents are called workbooks. Every workbook has sheets, usually called spreadsheets. You can add as many sheets as you want to a workbook or create new workbooks to keep the data separate (https://support.microsoft.com/ro-ro/).

RESULTS AND DISCUSSIONS

1. IDENTIFICATION OF WORK STAGE WITH AUTOMATION POTENTIAL

1.1 Criteria required for automation

This paper presents the working method used on a general cadastre project closely following each stage, in order to determine the possibility of their automation. To achieve this, the following criteria will be taken into account:

- The degree of repeatability of the steps that make up the stage
- The difficulty of producing the stage manually
- Time needed to complete the manual stage
- Possibility of automation

The stage of introducing the information identified and pre-used for parcelling into specific strata fulfils all the criteria mentioned above. For each building, it will be necessary to manually enter each information on layers as follows: property title, surface area, parcel, number, plot number, use category, document area, measured area and land book number. This stage requires an increased degree of attention from the person designated for its realization, but also a high working time that can be reduced with the help of implementing an automated application.

	A	B	E	F	G	H	I	J	K
1	Nr. Crt.	Titlu de proprietate	Număr tarla	Număr parcela	Categorie de folosință	Suprafață act	Suprafață măsurată	Număr carte funciară	Proprietar
2	1	2/9	58/2	1	Arabil	8700	8700		BXX XXXXA
3	2		58/2	2	Arabil	2900	2900		REZERVA
4	3	1/43	58/2	3	Arabil	7200	7200		BXXXX XXXXF
5	4	1/55	58/2	4	Arabil	7200	7200		XXXRU XXXXN
6	5	1/84	58/2	5	Arabil	7100	7100		MXXXXXCI FXXXXX
7	6	647302/54	58/2	6	Arabil	17100	17100		XXXXBU XXXXXXXN
8	7	28444/53	58/2	7	Arabil	11400	11400		SXXXXXXX DXXXX
9	8	2/25	58/2	8	Arabil	5700	5700		BXXX XXXXN
10	9	1/44	58/2	9	Arabil	2900	2900	402879	XXXXXXE XXXXXU-SXXXXXX, DXXXXX XXXXXA-AXXXX
11	10		58/2	10	Arabil	5800	5800		REZERVA

Figure 1. Registrysurface area 58/2 [source: Author]

1.2 Description of selected stage

The chosen step will be called text insertion in DWG, where we will take as an example parcel 58/2 within sector 16 of the Coșteiu territorial administrative unit, where the identification of thesurface area and parcels was carried out and the information thus obtained was entered into Excel and saved under the name of parcel register 58/2, as can be seen in figure 1.

The parcelling was carried out according to the register in Figure 1 and each resulting parcel was given the plot number, as can be seen in Figure 2, and then the other necessary information was entered.

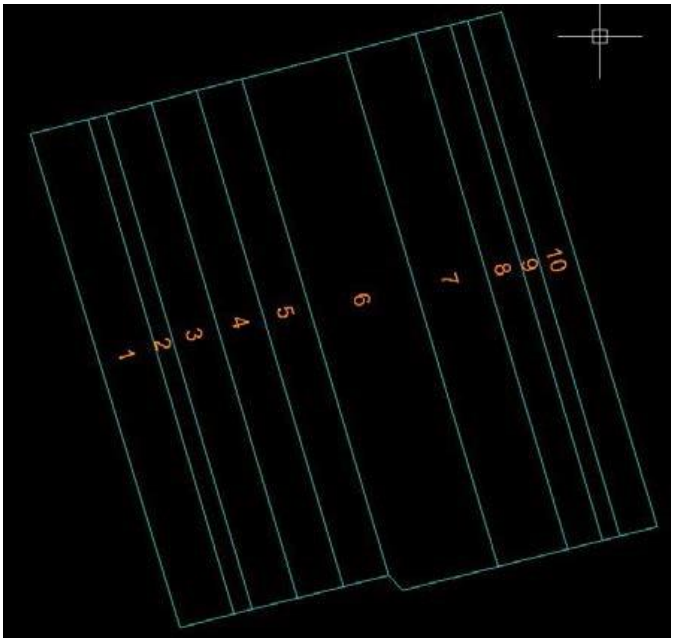

Figure 2. Surface area parcelling 58/2 [source: Author]

After completing the parcelling phase, the layers with a specific name for each type of information found in the surface area register in figure 3.1 will be created. They will be needed for the next stage, as you can see in figure 3.

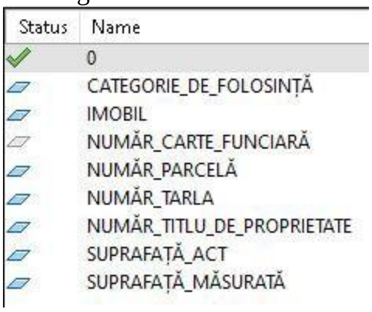

Figure 3. Layers created according to the registry [source: Author]

With the layers created, we can enter the information from the register in line for each plot, and then manually position the text inside the plot ordered by its position in the table head of the register from left to right, as can be seen in figure 4.

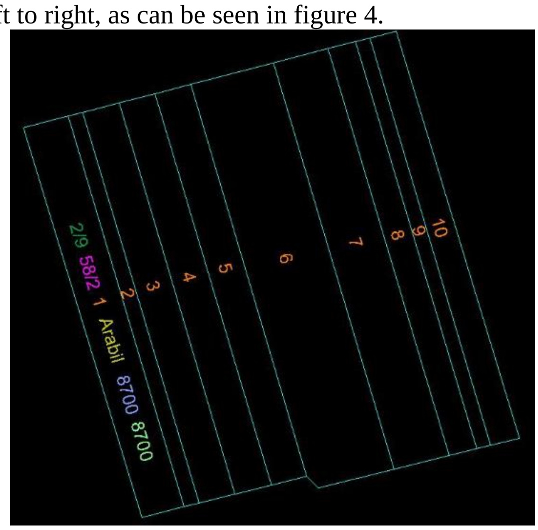

Figure 4. Manual input of information by layers [source: Author]

In this example we have only 10 plots that will be populated with the information from the register shown in figure 1 to reach the end of the current stage. We can also have a surface area with 100 or more plots where also the time required to complete the stage will be directly proportional to the number of plots. Most of the time this stage is executed on a cadastral sector, where in most cases several surface areas can be found.

1.3 The role of the selected stage

Because the generation of CGXML files will be done with the help of another software specialized in managing and populating databases (HERBEI & SALA, 2016) with the help of elements that can be inserted into DWG files, it is very important that the text insertion step in DWG is performed correctly. That is why we need the information from the created registry to be entered correctly on the specific layer, inside the assigned plot precisely so as not to mis populate the database that will be created in the next stage.

Any human error can lead to one or more CGXML files with wrong information. In order to verify the correctness of the execution at the completion of the text insertion stage in DWG, most of the time we will need a similar duration of time to that of execution.

By automating this stage, we will aim to reduce the necessary execution time and eliminate human errors, thus ensuring that the database will be populated with the correct information and corresponding strictly to the parcels listed in the surface area register.

2. WORK STEPS FOR AUTOMATION

2.1. Understanding the problem

In order to be able to start planning the automation process of the text insertion stage in DWG, it is necessary to know the execution steps and understand the commands used in AutoCAD to achieve them.

The stage is carried out in the following way:

- Create layers, where we use the Layer command accompanied by the New Layer function to create all the necessary layers;
- Insert Text, where we use the Text command to insert a single line of text on the currently selected layer;
- Text alignment and positioning, where we use the Move command to move the text in the desired plot but also the Rotate command to rotate the text according to the orientation position of the plot.

2.2. Drawing up the automation plan

The automation plan (POPESCU ET ALL, 2016, SMULEAC ET AL, 2012) will be fragmented into two distinct parts:

- Development of automation plan, where we will specify the necessary functions to make automation possible;
- Development of automation plan solution, where the approach of an efficient automation solution will be presented;

2.2.1. Development of automation plan

In order to be able to develop the automation plan (figure 5), it will be necessary to analyse the already existing elements without which this stage could not be executed. This will be the early phase of developing the automation plan.

The individual outline of plots made with closed polyline helps us to accurately locate the plot and can play an important role in approaching a solution for automatic arrangement of text inside it.

The fact that we already have the information from the surface area register entered inside the parcel, on the NUMĂR_PARCELĂ layer, will help us find an easier to implement solution for automatically inserting the other text information inside the parcel, spending less time solving this problem.

The existence of the surface area register, which includes the information intended for insertion inside the plot, facilitates the significant reduction of human errors of incorrect insertion of the text both in terms of its correctness of writing and placement on the incorrect layer, which also leads to incorrect retrieval of data in the database to be created in the next stage of execution of the general cadastre work.

Once the analysis phase of existing elements is completed, we will be able to focus on developing the automation plan in small steps. It will be composed of clear ideas, which together will form the mode of operation behind the automation process. For easy further writing of code in the AutoLISP programming language, we will pass these ideas as comments inside the file where we will develop new commands that will make automation itself possible. In the AutoLISP programming language, any character written after a semicolon on that line of code will be ignored and will not be executed as part of a command or function. This is called a comment by programmers and is usually used for situations where the code will be reverted later and facilitates its understanding if it has been described correctly and intuitively.

The automation will be divided into steps as follows:

- Step 1 – The coordinates and text of the elements located on the NUMĂR_PARCELĂ layer will be exported in textual format, so we will know exactly the coordinates in which we will insert the information from the surface area register and we will associate the information correctly according to the plot number.
- Step 2 – Prepare in Excel a formula for unifying the information among each parcel entered in thesurface area register so that for each piece of information we have a special character that will make it possible to sort by layers automatically in the following steps, but each information will be separated by a space.
- Step 3 – We will insert in Excel a command to insert line of text for the coordinates exported in step 1 with the text obtained at the end of step 2 for each plot. This set of commands will be entered directly into the command line in AutoCAD.
- Step 4 – After inserting the text from the previous step, we will use a command to align it based on a line drawn through two specified points, preferably a line of the outline of the plot.
- Step 5 – We will use a command to divide the text from the line of text into words having as reference the spacing between the words imported in step 3.
- Step 6 – Develop a command that will create the necessary layers and place each text on the corresponding layer having as differentiation system the existence of a special character for each type of information specified in step 2.
- Step 7 – Delete special characters from text using AutoCAD function followed by replace function (*eng. replace*) with nothing.

VSC_Lisp > Automatizare.lsp

```
;Pasul 1 - Exportare în format de tip text al coordonatelor tuturor elementelor/
;/de pe stratul NUMĂR_PARCELĂ inclusiv textul regăsit pe strat.
;Pasul 2 - Pregătirea în excel a unei formule pentru unificarea informațiilor/
;/din celulele de pe aceași parcelă cu specificația de a avea pentru fiecare/
;/tip de informație în parte un caracter special destinat pentru a știi pe ce strat/
;/va fi mutat ulterior în autocad.
;Pasul 3 - Pregătirea unei comenzi de inserare a textului pregătit la -pasul 2-/
;/în excel pentru fiecare parcelă în coordonatele exportate la -pasul 1-, acest/
;/set de comenzi va fi introdus în linia de comanda, nu în LISP.
;Pasul 4 - Inserarea unei comenzi existente pentru alinierea textului inserat /
;/la -pasul 3- având ca bază o linie trasată prin doua puncte.
;Pasul 5 - Inserarea unei comenzi existente pentru divizarea textului având ca/
;/reper spațierea dintre cuvintele importate la -pasul 3-.
;Pasul 6 - Dezvoltarea unei comenzi ce va creea automat straturile necesare/
;/și va plasa fiecare text pe stratul corespunzător caracterului special /
;/desemnat la -pasul 2-.
;Pasul 7 - Înlăturarea caracterelor speciale din text folosind funcția find/
;/din autocad urmată de replace cu -nimic-.
```

Figure 5. Automation plan commented [source: Author]

2.2.2. Automation plan resolution mode

We will make an analysis of the steps elaborated in the automation plan (figure 5) in order to identify the steps that can be solved with the help of already existing AutoLISP applications, created by other AutoCAD users and posted on various specialized websites, all this to reduce the working time for implementing the automation process.

As regards the information entered in the parcel register, the Excel application will be used in which the concatenate function will be used to join the text strings in the row of each parcel, thus forming a whole string of information for the plot in question, but also to write correctly and quickly the commands for inserting text in coordinates.

For steps that do not have an AutoLISP application developed, we will manually perform the working mode for a better understanding of the commands and functions applied in its realization.

2.3. Materialisation of the resolution plan

Once we think about how to solve it, we will start writing in the commented .lsp file the code necessary for each mentioned step, marking the beginning and end of each step, where appropriate, and for the part where we help ourselves with Excel we will add comments if necessary.

The materialization of the settlement plan will be done in two parts:

- AutoLISP development part, where reference will be made to how to implement existing functions and customize them according to our strict needs in this automation project for the text insertion stage in DWG.
- The Microsoft Excel use part, where we will fix our attention on how we can help ourselves by the Excel application to simplify the solution of the automation plan using a few key functions.

2.3.1 AutoLISP development part

The Visual Studio Code source code editor (figure 6) will be used, which is easy to use and contains many helpful extensions, including AutoCAD, AutoLISP Extension and Rainbow Brackets, used to assist in writing code and help to visualize written code more easily.

A file will be created within the editor containing the source code of all integrated or developed commands. In this file we will have as a first step the written comments when developing the automation plan, which are used as a guide for integrating and developing orders in the correct order. The lines of code for each step will be passed between two comments that will mark and suggest the beginning and end of the current step. After the opening comment we will have the comment that will describe the functions performed by the inserted code. In figure 7 we can easily distinguish comments, they are written on each line after the semicolon sign where the text colour has changed to green, this being preset from the installation of AutoCAD AutoLISP Extension.

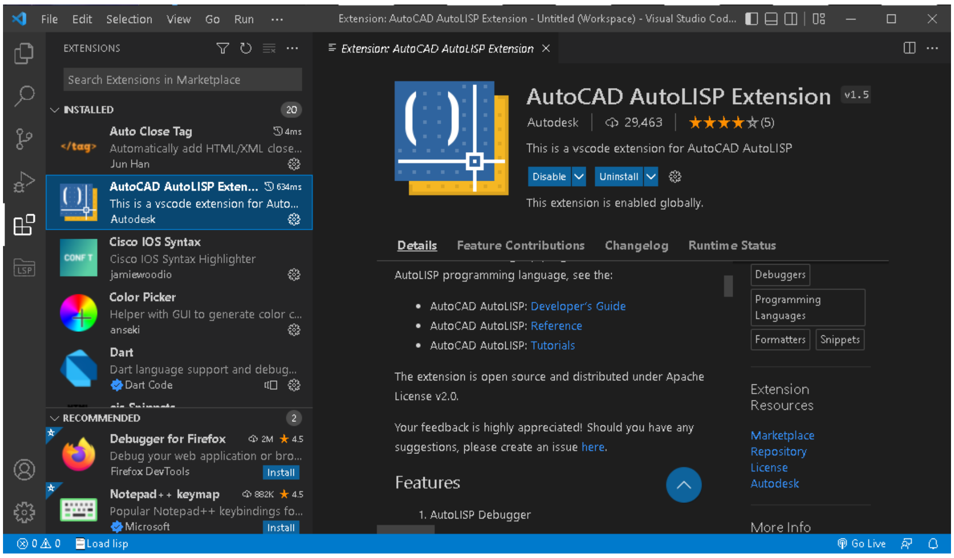

Figure 6. Visual Studio Code source code editing software [source: Author]

```
VSC_Lisp > Ajutor.lsp
; Pasul 1 Început

;Pasul 1 - Exportare in format de tip text al coordonatelor tuturor elementelor/
;/de pe stratul NUMĂR_PARCELĂ inclusiv textul regăsit pe strat.

cod lisp

; Pasul 1 Final
```

Figure 7. Exemplifying the use of comments [source: Author]

```
(defun c:t2csv ( / ss fname obj objstr txtins x )
(setq ss (ssget '((0 . "*TEXT"))))
(if (/= ss nil)
  (progn
    (setq fname (open "D:\\acadtemp\\text2csv.txt" "w"))
    (repeat (setq x (sslength ss))
      (setq obj (vlax-ename->vla-object (ssname ss (setq x (- x 1)))))
      (setq objstr (vla-get-textstring  obj))
      (setq txtins (vlax-get obj 'InsertionPoint))
      (if (= (vla-get-objectname obj) "AcDbMText")
       (setq objstr (LM:UnFormat objstr nil))
      )
      (setq objstr (strcat (rtos (car txtins) 2 3) "," (rtos (cadr txtins) 2 3) "," (rtos (caddr txtins) 2 3) "," objstr))
      (write-line objstr fname)
    (princ (strcat "\n" (rtos x 2 0)))
    )
    (close fname)
  )
  (alert "No text selected try again")
)
(princ)
)
```

Figure 8. Value of the fname variable to be changed [source: Author]

After analysing the elaborated plan, the following aspects were noticed that clarify the way of working regarding the writing part of the code:

- For step 1 there is a developed application that we will integrate into our source code and only the value of the fname variable will change (figure 8), which defines the path to save the Excel file of type .csv that will contain the coordinates and text of the elements selected for export. Figure 8 shows the line on which the aforementioned variable will be changed. To use this function, after selecting items for export, the text will be written in the AutoCAD command line *ExportCSV* followed by the *Enter*, finally checking in the modified path of the code line, the existence of the newly created file and the validity of the information.
- In step 4 we will insert an application that meets the current needs for automation and was developed by other AutoCAD users years ago. This command will only have to be integrated into our source code with a different name. The command required to enter in the AutoCAD command line is named *textalign* and allows alignment of one or more selected texts after a line drawn by colons.
- Step 5 will make it possible to separate words from line of text by implementing another application developed by AutoCAD users. The command used in AutoCAD is called *wordsfromtext* and has the role of dividing the line of text into words, having as criterion the division of the space between texts.
- For step 6, an application will be developed that will create and change the layer according to the special characters that have been previously assigned for each type of information in the surface area register.

Step 6 will contain a series of functions that will take turns creating and moving texts with a certain special character written before the information in the surface area register, on the correct layer.

We will start by defining the first function that can be used in the command line under the name of ExchangeTcatfol.

Using IF syntax, we condition that the SS variable be composed of the selection of text or mtext elements containing the special character <^> that can be followed by any other text.

To create the layer with the name USE_ CATEGORY, we will use another conditioning performed with the IF syntax, as if you search in the LAYER table for this layer and will not be found to create the layer with the same name, having the base colour selected by the colour code true 255,223,127 and to be selected as the current layer.

For the next conditioning, set a variable KK equal to zero and use the while syntax to specify that as long as KK is less than SS, change the property of the selected elements in terms of the layer they are on with the layer named USE_ CATEGORY. In order not to create an infinite loop, at the end of this conditioning we will set the variable *KK* as equal to 1.

```
(defun C:SchimbTcatfol (/)
(if (setq SS (ssget "X" (list (cons 0 "TEXT,MTEXT") (cons 1 "^*"))))
(progn
(if (not (tblsearch "LAYER" "CATEGORIE_DE_FOLOSINTA")) (command "LAYER" "M" "CATEGORIE_DE_FOLOSINTA" "C" "T" "255,223,127" "" ""))
(setq KK 0)
(while (< KK (sslength SS))
(command "CHPROP" SS "" "LA" "CATEGORIE_DE_FOLOSINTA" "")
(setq KK (1+ KK))
)
)
(princ)
)
)
```

Figure 9. The source code USE_ CATEGORY layer [source: Author]

In figure 9 we have the source code for the first function from step 6 shown in the last 4 paragraphs. The lines of code were described in an orderly manner from top to bottom.

The procedure will be similar for the remaining layers, rewrite the code in figure 3.9 and change the layer name, special character and colour code of the layer one by one. We will have the following changes as follows:

- For layer NUMBER_LAND_REGISTRY the command C:ExchangeTnrCF will be defined, for which the special character < $ > and colour code 0.255,255 will be used;
- For layer NUMBER_PARCELL the command C:ExchangeTnrP will be defined, for which the special character < : > and colour code 255,127.0 will be used;
- For layer NUMBER_SURFACE AREA command C:ExchangeTnrT will be defined, for which the special character < ! > and colour code 255,0,255;
- For the NUMBER_PROPRIETY_TITLE layer will be defined the command C:ExchangeTnrTP for which the special character < will be used; > and colour code 0,127,0;
- For layer SURFACTE_DOCUMENT the command C:ExchangeTsA will be defined, for which the special character < ' > and colour code 102,204,102 will be used;
- For layer MEASURED_SURFACE the command C:ExchangeTsM will be defined, for which the special character < & > and colour code 102,153,204 will be used;

Once we have completed defining and writing all the commands mentioned above, we will define a C:ExchangeLayer command that will execute all commands from step 6 as shown in figure 10. This ensures that we will have the information passed on each destination layer.

```
(defun C:SchimbStrat ()
(C:SchimbTcatfol)
(C:SchimbTnrCF)
(C:SchimbTnrP)
(C:SchimbTnrT)
(C:SchimbTnrTP)
(C:SchimbTsA)
(C:SchimbTsM)
(princ)
)
```

Figure 10. Code written for executing all commands written in step 6 [source: Author]

2.3.2 Microsoft Excel User Part

Microsoft Excel is an application (CASIAN ET AL, 2019) that uses spreadsheets to organize numbers and dates using formulas and functions with different uses. That is why it will be used as an aid for the successful implementation of the developed automation plan.

Due to its ease of use, Excel is frequently used to shorten the time it takes to organize and write repetitive commands for AutoCAD and other programs.

Unifying information from the same plot on a single text string and adding special characters to divide information on the intended layers will be possible using the CONCATENATE function. The surface area register being created in the excel application will ease our work, already having the information structured in columns, and the information in each row belongs to a plot in that surface area.

Since you can see cells that do not contain information and do not have a constant, we will use the IF and ISBLANK functions alongside the CONCATENATE function. To achieve the desired result, we will use the mentioned functions in the following way:

=CONCATENATE(IF(ISBLANK(A11),"",";"   &A11),"   ",IF(ISBLANK(D11)), where:

CONCATENATE is the function that makes it possible to join information from two or more cells, IF together with ISBLANK make it a condition that when the cell is without information to add to the string of united information found in the expression < """> meaning nothing, and if the cell contains information it will join it the special character < ; > .

The final formula for the table in figure 11 will be as follows:

=CONCATENATE(IF(ISBLANK(A11),"",";" &A11),"",IF(ISBLANK(D11),"","!" &D11),"",IF(ISBLANK(E11),"",":"&E11),"",IF(ISBLANK(F11),"","^"&F11),"",IF(ISBLANK(G11),"",","&G11),"",IF(ISBLANK(H11),"","&"&H11),"",IF(ISBLANK(I11),"","$"&I11))

=CONCATENATE(IF(ISBLANK(A2),"",";"&A2)," ",IF(ISBLANK(D2),"","!"&D2)," ",IF(ISBLANK(E2),"",":"&E2)," ",IF(ISBLANK(F2),"","^"&F2)," ",IF(ISBLANK(G2),"","'"&G2)," ",IF(ISBLANK(H2),"","&"&H2)," ",IF(ISBLANK(I2),"","$"&I2))

	A	D	E	F	G	H	I	J
1	Titlu de proprietate	Număr tarla	Număr parcela	Categorie de folosință	Suprafață act	Suprafață măsurată	Număr carte funciară	
2	2/9	58/2	1	Arabil	8700	8700		;2/9 !58/2 :1 ^Arabil '8700 &8700
3		58/2	2	Arabil	2900	2900		
4	1/43	58/2	3	Arabil	7200	7200		
5	1/55	58/2	4	Arabil	7200	7200		
6	1/84	58/2	5	Arabil	7100	7100		
7	647302/54	58/2	6	Arabil	17100	17100		
8	28444/53	58/2	7	Arabil	11400	11400		
9	2/25	58/2	8	Arabil	5700	5700		
10	1/44	58/2	9	Arabil	2900	2900	402879	
11		58/2	10	Arabil	5800	5800		

Figure 11. Register of the surface area with the formula CONCATENATE applied [source: Author]

=CONCATENATE(IF(ISBLANK(A2),"",";"&A2)," ",IF(ISBLANK(D2),"","!"&D2)," ",IF(ISBLANK(E2),"",":"&E2)," ",IF(ISBLANK(F2),"","^"&F2)," ",IF(ISBLANK(G2),"","'"&G2)," ",IF(ISBLANK(H2),"","&"&H2)," ",IF(ISBLANK(I2),"","$"&I2))

	A	D	E	F	G	H	I	J	K
1	Titlu de proprietate	Număr tarla	Număr parcela	Categorie de folosință	Suprafață act	Suprafață măsurată	Număr carte funciară		
2	2/9	58/2	1	Arabil	8700	8700		;2/9 !58/2 :1 ^Arabil '8700 &8700	
3		58/2	2	Arabil	2900	2900		!58/2 :2 ^Arabil '2900 &2900	
4	1/43	58/2	3	Arabil	7200	7200		;1/43 !58/2 :3 ^Arabil '7200 &7200	
5	1/55	58/2	4	Arabil	7200	7200		;1/55 !58/2 :4 ^Arabil '7200 &7200	
6	1/84	58/2	5	Arabil	7100	7100		;1/84 !58/2 :5 ^Arabil '7100 &7100	
7	647302/54	58/2	6	Arabil	17100	17100		;647302/54 !58/2 :6 ^Arabil '17100 &17100	
8	28444/53	58/2	7	Arabil	11400	11400		;28444/53 !58/2 :7 ^Arabil '11400 &11400	
9	2/25	58/2	8	Arabil	5700	5700		;2/25 !58/2 :8 ^Arabil '5700 &5700	
10	1/44	58/2	9	Arabil	2900	2900	402879	;1/44 !58/2 :9 ^Arabil '2900 &2900 $402879	
11		58/2	10	Arabil	5800	5800		!58/2 :10 ^Arabil '5800 &5800	
12									

Figure 12. Register of the surface area with the Concatenate formula applied for all plots [source: Author]

Once the formula for a row is created, it can be easily and mistakenly multiplied for each cell in the same column using the copy setting from Autofill Options as can be seen in figure 12.

Starting from the command (command "_TEXT", "J", "C", "X ,Y", "2", "100", "my text"), required for step 3, you will obtain, with the help of an Excel spreadsheet, a formula in which you can easily insert the text and coordinates needed to add in DWG. As can be seen, the command is used _TEXT with the justify functions followed by *centre* selection, X, Y coordinates, text height of 2, oblique text noise of 100 and finally the desired text (figure 13).

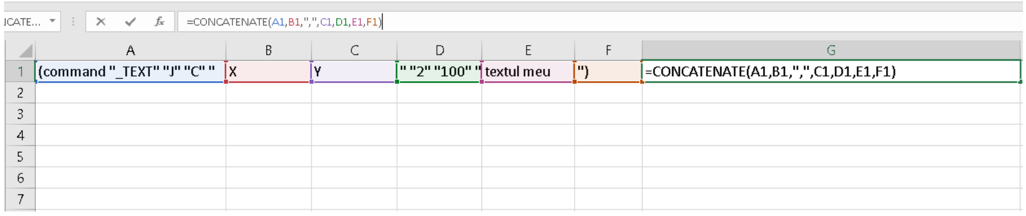

Figure 13. Text insertion command fragmentation and CONCATENATE formula used [source: Author]

The CONCATENATE formula that will be used to automate text insertion commands is the following: =CONCATENATE(A1,B1,",",C1,D1,E1,F1), where we regroup the command after entering the real X, Y coordinates but also the text to be inserted, and between the coordinates we put < , > to select the insertion point in AutoCAD.

3.3 APPLICATION OF THE DEVELOPED AUTOMATION PLAN

After solving and developing the automation application, we will open the DWG in which a parcelled surface area is found, then load the .lsp file into AutoCAD using the *APPLOAD command,* from where we will search for the AutoLISP application path and click on the *Load* button, then on the *Close* button, as can be seen in figure 14.

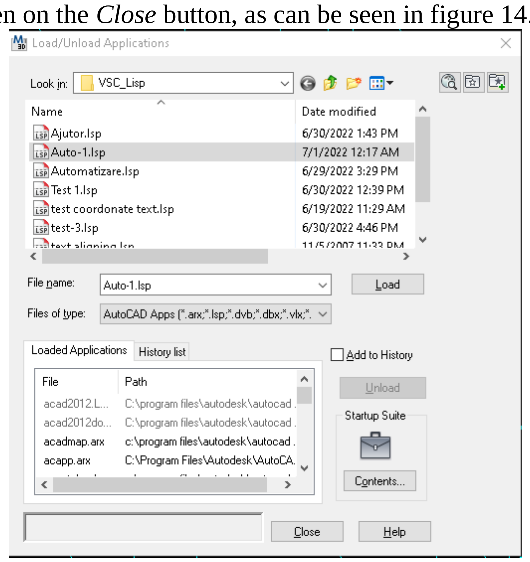

Figure 14. Loading AutoLISP into AutoCAD using the APPLOAD command [source: Author]

Once the application has been loaded, the steps will be followed as planned:

Step 1 – Select all elements on the NRPARCELATP layer using the select function and use the EXPORTCSV command according to figure 15.

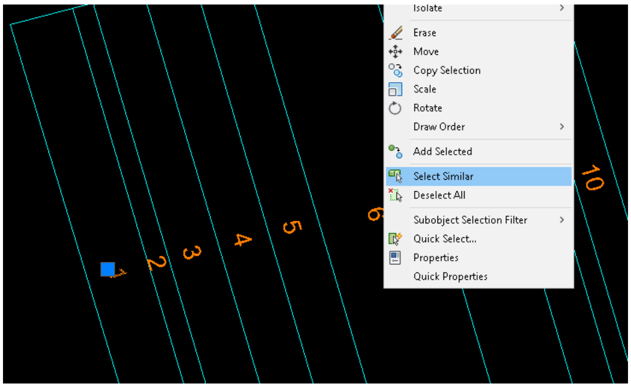

Figure 15. Use similar select function [source: Author]

Step 2 – In the surface area register, apply the formula CONCATENATE written to the usage part of Microsoft Excel to unify all the information on the row of each parcel.

Step 3 – Delete all text elements from DWG, then open the file exported in step 1, figure 16, to correctly write the coordinates necessary to insert the text obtained in step 2 inside the plots but also the Excel file with the formula for completing the text insertion command, figure 17, and copy and paste directly into AutoCAD as can be seen in figure 18.

	A	B	C	D	E	F	G	H	I
1	254197.777	480430.342	0	1					
2	254217.640	480435.917	0	2					
3	254234.941	480441.133	0	3					
4	254259.810	480447.458	0	4					
5	254284.559	480453.472	0	5					
6	254326.432	480460.002	0	6					
7	254374.191	480470.944	0	7					
8	254403.106	480476.401	0	8					
9	254416.201	480484.007	0	9					
10	254432.348	480481.689	0	10					

Figure 16. Excel CSV file with data exported in step 1 [source: Author]

	A	B	C	D	E	F	G
1	(command "_TEXT" "J" "C" "	254197.7770	480430.3420	" "2" "100"	;2/9 !58/2 :1 ^Arabil '8700 &8700	")	(command "_TEXT" "J" "C" "254197.777,480430.342" "2" "100" ";2/9 !58/2 :1 ^Arabil '8700 &8700 ")
2	(command "_TEXT" "J" "C" "	254217.6400	480435.9170	" "2" "100"	!58/2 :2 ^Arabil '2900 &2900	")	(command "_TEXT" "J" "C" "254217.64,480435.917" "2" "100" " !58/2 :2 ^Arabil '2900 &2900 ")
3	(command "_TEXT" "J" "C" "	254234.9410	480441.1330	" "2" "100"	;1/43 !58/2 :3 ^Arabil '7200 &7200	")	(command "_TEXT" "J" "C" "254234.941,480441.133" "2" "100" ";1/43 !58/2 :3 ^Arabil '7200 &7200 ")
4	(command "_TEXT" "J" "C" "	254259.8100	480447.4580	" "2" "100"	;1/55 !58/2 :4 ^Arabil '7200 &7200	")	(command "_TEXT" "J" "C" "254259.81,480447.458" "2" "100" ";1/55 !58/2 :4 ^Arabil '7200 &7200 ")
5	(command "_TEXT" "J" "C" "	254284.5590	480453.4720	" "2" "100"	;1/84 !58/2 :5 ^Arabil '7100 &7100	")	(command "_TEXT" "J" "C" "254284.559,480453.472" "2" "100" ";1/84 !58/2 :5 ^Arabil '7100 &7100 ")
6	(command "_TEXT" "J" "C" "	254326.4320	480460.0020	" "2" "100"	;647302/54 !58/2 :6 ^Arabil '17100 &17100	")	(command "_TEXT" "J" "C" "254326.432,480460.002" "2" "100" ";647302/54 !58/2 :6 ^Arabil '17100 &17100 "
7	(command "_TEXT" "J" "C" "	254374.1910	480470.9440	" "2" "100"	;28444/53 !58/2 :7 ^Arabil '11400 &11400	")	(command "_TEXT" "J" "C" "254374.191,480470.944" "2" "100" ";28444/53 !58/2 :7 ^Arabil '11400 &11400 ")
8	(command "_TEXT" "J" "C" "	254403.1060	480476.4010	" "2" "100"	;2/25 !58/2 :8 ^Arabil '5700 &5700	")	(command "_TEXT" "J" "C" "254403.106,480476.401" "2" "100" ";2/25 !58/2 :8 ^Arabil '5700 &5700 ")
9	(command "_TEXT" "J" "C" "	254416.2010	480484.0070	" "2" "100"	;1/44 !58/2 :9 ^Arabil '2900 &2900 $402879	")	(command "_TEXT" "J" "C" "254416.201,480484.007" "2" "100" ";1/44 !58/2 :9 ^Arabil '2900 &2900 $402879"
10	(command "_TEXT" "J" "C" "	254432.3480	480481.6890	" "2" "100"	!58/2 :10 ^Arabil '5800 &5800	")	(command "_TEXT" "J" "C" "254432.348,480481.689" "2" "100" " !58/2 :10 ^Arabil '5800 &5800 ")

Figure 17. Command in automated coordinate text setting [source: Author]

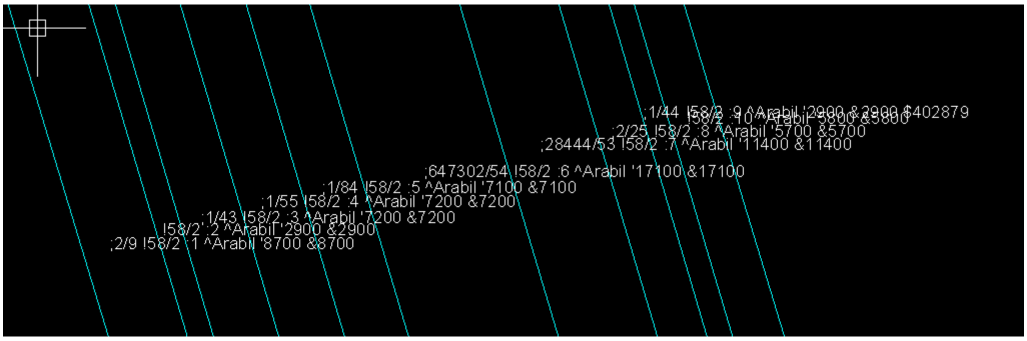

Figure 18. Text string with special characters inserted in DWG [source: Author]

Step 4 – After selecting all text elements in DWG using similar select, use the *textaligne* function, and align the text according to a drawn line, figure 19.

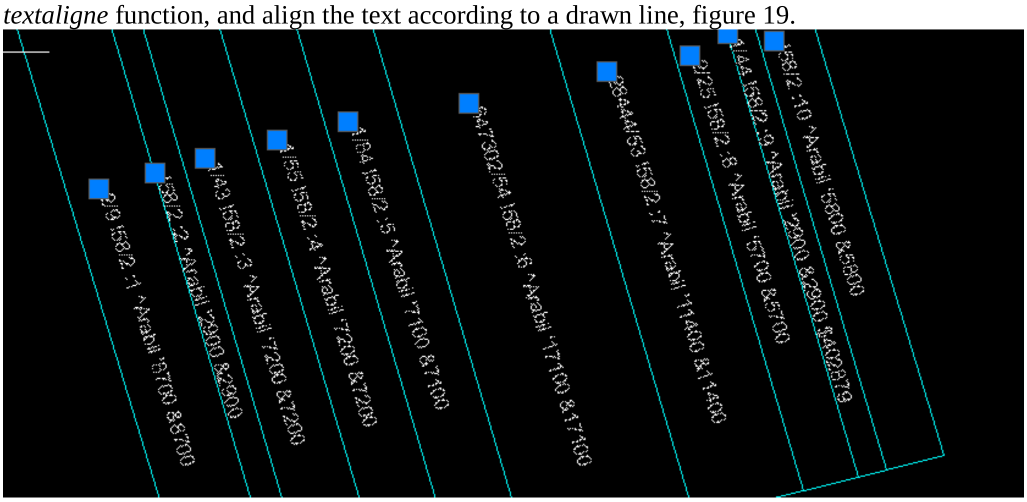
Figure 19. Text aligned after plot outline [source: Author]

Step 5 – The t2w command will be used to divide the text string by the space between words.

Step 6 – To change the layer of each text that was separated in step 5, the ExchangeLayer command will be used, which will have the role of doing this automatically.

Step 7 – After we have successfully completed step 6, the *find* and *replace* function will be used for special characters, in DWG only the information from the surface areal register remains, as can be seen in figure 20.

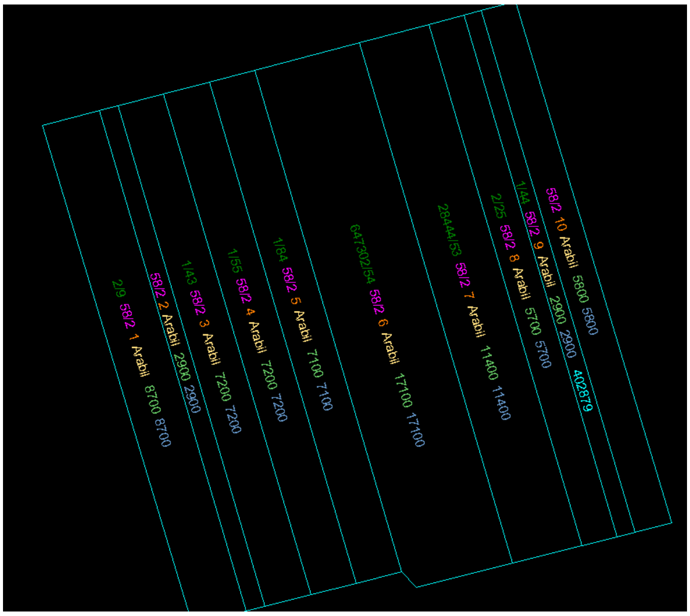

Figure 20. Final result after automation [source: Author]

CONCLUSIONS

At this point, we can say about the implementation of automation that it is a process that can consume more time than performing the stage of inserting text into DWG manually, but after materializing the automation plan, it can be used for countless projects, gaining time when automation is applied to an entire cadastral sector.

For a small surface area, the time required to implement the developed automation plan may be similar to the time to achieve it manually, but as the number of plots increases, the implementation time of the automation plan will be lower compared to that of performing the manual stage.

Another important aspect to mention is that this automation plan is not complete, it can be constantly improved with new ideas, but only if it is applied constantly will you be able to notice those small details that can bring extra time saved in the execution of that stage.

By resuming the resolution plan, we can also observe other ways of approaching the current solution and reach a faster improvement than if we observed the way the plan works in execution.

The human errors that could have occurred during the execution of this stage have been mostly eliminated with the new automated way of working, which gives us confidence that the final product of this work is a clear and correct one.

BIBLIOGRAPHY

AutoLISP Reference Guide, AutoCAD 2012 for MAC, 2011 Autodesk, Inc. All Rights Reserved
Badea Gh., Cadastre, 2013, Conspress Publishing House, Bucharest
Bos N., Iacobescu O., 2009, Cadastre and Land Book, C.H. Beck Publishing House, Bucharest
Casian, A., Şmuleac, A., & Simon, M. (2019). Possibilities of using the UAV photogrammetry in the realization of the topo-cadastral documentation

CUCOȘ CONSTANTIN, 2023, Education and temporality, https://doxologia.ro/educatie-temporalitate

DECREE-LAW NO. 115 OF APRIL 27, 1938 for the unification of provisions regarding land books, Published in the Official Gazette No. 95 of April 27, 1938, https://legislatie.just.ro/Public/DetaliiDocument/25

FAINITA GABRIELA, 2016, Land Registry Overview, https://projurista.ro/cartea-funciara-prezentare-generala/

GREEN R., 2008, The AutoLisp Crash Course,Autodesk

HERBEI, M., & SALA, F. (2016). Classification of land and crops based on satellite images Landsat 8: case study SD Timisoara

HTTPS://CODE.VISUALSTUDIO.COM/, ACCESSED ON 13.04.2023

HTTPS://CODE.VISUALSTUDIO.COM/DOCS/EDITOR/INTELLISENSE, ACCESSED ON 01.04.2023

HTTPS://DOCS.MICROSOFT.COM/RO-RO/POWER-APPS/MAKER/PORTALS/VS-CODE-EXTENSION, ACCESSED ON 01.04.2023

HTTPS://SUPPORT.MICROSOFT.COM/RO-RO/, ACCESSED ON 01.04.2023

HTTPS://WWW.ITLEARNING.RO/CAPITOLUL-1-PREZENTARE-GENERALA-A-PROGRAMULUI-AUTOCAD/, ACCESSED ON 12.04.2023

MARGARET ROUSE, 2015, Intellisense. What Does Intellisense Mean?, https://www.techopedia.com/definition/24580/intellisense

OPREA LUCIANA ET AL., 2013, "Introduction of the general cadastre in an administrative territory", http://pangeea.uab.ro/upload/20_261_7_oprea_-_introducerea_cadastrului.pdf

PLAN A PROJECT WITH A WORK BREAKDOWN STRUCTURE (PROJECT SERVICE), 2023, HTTPS://LEARN.MICROSOFT.COM/RO-RO/DYNAMICS365/PROJECT-OPERATIONS/PSA/SCHEDULE-PROJECT-WORK-BREAKDOWN-STRUCTURE

POPESCU, G., POPESCU, C. A., HERBEI, M., & SMULEAC, A. (2016). Measuring the parameters that influence the phenomenon of displacement and deformation of the ground at Mina Livezeni

SABĂU, N., CRAINIC, G., 2006, Applications of remote sensing in cadastre, University of Oradea Publishing House

SELF-DISCIPLINE AND MANAGEMENT OF THE MOST PRECIOUS RESOURCE – TIME, HTTPS://THEWOMAN.RO/AUTODISCIPLINA-SI-MANAGEMENTUL-CELEI-MAI-PRETIOASE-RESURSE-TIMPUL/

SMULEAC, A., HERBEI, M., & POPESCU, C. (2012). Creating the digital terrain model of the Usamvb Area using modern technology