

ANALYSIS OF APPROACHES TO CUSTOMISATION OF AUTODESK-PRODUCTS

Svetlana Toncheva

University of Mining and Geology "St. Ivan Rilski", 1700 Sofia; svetlana.toncheva@mgu.bg

ABSTRACT. Some of the most popular CAD-systems worldwide are the products by the Autodesk company, and AutoCAD in particular. One of the reasons for this popularity is the variety of tools and methods for customising the product. This article is a review and an analysis of the methods and techniques for the customisation of Autodesk products. The focus lies mainly on the programming languages which can be used for this purpose.

Keywords: AutoCAD, custom settings, programming languages, shape, hatch pattern

АНАЛИЗ НА ПОДХОДИ ЗА ПОТРЕБИТЕЛСКИ НАСТРОЙКИ НА СРЕДАТА В AUTODESK-ПРОДУКТИ

Светлана Тончева

Минно-геоложки университет "Св. Иван Рилски", 1700 София

РЕЗЮМЕ. Едни от най-разпространените в света CAD-системи са продуктите на фирма Autodesk, в частност AutoCAD. До известна степен това се дължи на факта, че за тези продукти има развити средства и методи за тяхната потребителска настройка. В настоящата статия се прави обзор и анализ на начините и техниките за потребителски настройки на средата на Autodesk-продукти, като се акцентира главно върху езиците за програмиране, които могат да се ползват за целта.

Ключови думи: AutoCAD, потребителски настройки, езици за програмиране, шейпове, шаблони за шриховане

Introduction

The Autodesk Company is undeniably a world leader in producing graphic software. Its CAD-systems are some of the most commonly used worldwide. There are probably many reasons for this popularity, however one of them certainly is the ability to customise the products' settings and to extend them with additional functionality, in order for specific custom tasks to be solved. Autodesk provides various tools for this purpose.

Custom settings are useful in two aspects:

- They make work faster and easier for every single person;
- They facilitate team work when unifying the system settings for all participants in a given project.

Therefore, it is useful for users to be familiar with the customisation options.

The Autodesk company itself suggests 8 Top Ways to Customise AutoCAD (Top Ways to Customise AutoCAD, 2019):

- *Exchange Apps.* The Exchange Apps marketplace includes general productivity tools, drawing content, and applications created by third party publishers for specific disciplines. The apps are easy to download and install.
- *Scripts.* Script files are ASCII text files with the file extension .scr and can be edited using a simple text editor such as NotePad on Windows or TextEdit on Mac OS. If you can type commands and options at the command line, you can write a script file.

- *Action Recorder.* Use the Action Recorder to record commands and input values that you can play back as an action macro to automate repetitive tasks.
- *Command Aliases.* Command alias are shortened command names that can be entered at the Command prompt as an alternative to the standard full command name and are stored in a program parameter PGP-file.
- *Dynamic Blocks.* Dynamic blocks contain rules, or parameters, for how to change the appearance of the block reference when it is inserted in the drawing.
- *Customisable and expandable ribbon UI.* Customise and expand the ribbon tabs, panels, and tools to suit your needs and meet your company's standards. It's easy to do in the Customise User Interface (CUI) dialog box.
- *Profiles.* Profile settings can include information such as default search and project file paths, template file locations, default linetype and hatch patterns, and printer defaults.
- *Plotter Configurations and Plot Styles.* Create plot configurations to configure your output devices for publishing or plotting drawings. Use custom plot styles to specify how object properties such as colour and lineweight are represented when plotted.

Custom settings can be categorised in two main groups:

1. Settings in command CONFIG, changes in GUI and setting suitable values for the system variables in AutoCAD.
2. Creating new types of objects.

What is characteristic for the first group is that the settings are set mainly in a dialog mode. They are relatively well documented and are not subject of the current article.

The second group is more interesting, since the creation of new objects can lead to automating some elements of designing and thus increase effectiveness of the work. The current article introduces these customisation aspects.

To improve readability, from now on the term „Autodesk-products” will be replaced with AutoCAD. This is the first product of Autodesk and one of the most popular CAD-systems for universal use worldwide. Everything presented in this article is valid not only for AutoCAD, but also for the other products of Autodesk that allow programming. The light ("LT") versions of the products make an exception, since they do not provide programming options except for the language DIESEL.

Creating new object

The AutoCAD user is given the opportunity to create new objects like line types, hatch patterns, menus (changes in GUI of the system generally), shapes, additional commands. This article will not treat the changes in GUI of the system since there are suitable means this to be done. Creation of blocks, dynamic blocks and blocks with attributes will not be commented here either, since it is given in almost each AutoCAD course and is described in many books. Creation of libraries with graphic symbols can be found in (Guneva, Trifonova, 2001).

Autodesk offers a couple of programming languages for creation of new commands.

The definitions of new line types, hatch patterns and shapes are to be fixed in external text files under a defined format. Three commands: MkLtype, MkShape and SuperHatch (Fig. 1a, b) have been added in "Express Tools" module since 2010. The first two of them generate automatically the text files defining line types and shapes, respectively, and the third one hatches a given closed area without creating any file defining the hatch pattern.

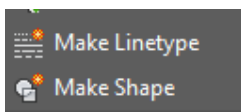


Fig. 1a.

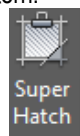


Fig. 1b.

"Express Tools" represent a set of functions, developed by Autodesk fans. The company spreads them under the explicit condition that it bears no responsibility for their good performance. "Express Tools" commands are available only if it is indicated explicitly in the course of AutoCAD installation that they should be installed.

It is a good idea Autodesk App Store to be checked for the available elements before undertaking creation of new objects. If the needed objects are available in Autodesk App Store they can be downloaded from there.

Creating shapes

Shapes are objects similar to blocks. Shapes are much more effective in the aspect of speed of visualisation and memory needed for saving, compared to the blocks. However, nowadays shapes are used only for the creation of complex types of lines due to the obstacles and limitations going along with their creation.

Contemporary versions of AutoCAD offer two ways for creation of shapes.

- using MkShape command (if Express tools is installed);
- manual creation of text file with extension shp.

The first approach is incomparably easier. It needs the shape image to be drawn first on the screen using AutoCAD commands then MkShape generates the shp-file and automatically compiles it to shx-file that AutoCAD is to use. Neither knowledge about the shapes description rules nor acquaintance with source shape files structures are needed. However, sometimes the results provided by MkShape are unsatisfying in the aspect of visualisation and are extremely ineffective. For example, to create a shape named Z2, representing a circle with a radius 1, MkShape generates the following content:

```
*1,58,Z2
4,101,4,99,3,125,3,5,3,64,002,9,(4,0),(0,0),001,9,(-1,2),
(-2,2),(-2,0),(-2,-2),(-1,-2),(1,-2),(2,-2),(2,0),(2,2),(1,2),(0,0),
002,9,(-4,0),(0,0),001,4,64,4,5,4,125,3,99,3,101,0
```

While it may be defined as follows:

```
*2,7,Z2
2,010,1,10,(1,000),0
```

Both definitions of the shape Z2 generate all the same image – a circle with a radius 1 and base point - the centre of the circle. In principle, the effectiveness of the visualisation depends on the count of the numbers in the description of the shape. That is why it is desirable this count to be minimised. In the example above the shape Z2 description contains 7 numbers in the case of the manual generation and 58 - in the case of the automated one.

On Figure 2(a, b) the visualisation of this shape is shown, resulting from the first and the second definitions, respectively.

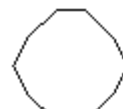


Fig. 2a.



Fig. 2b.

The example above shows, that the command MKShape (when available) needs to be used very carefully. Its application has to be reduced to private needs. If it is necessary to create shapes for other clients, it is recommended to take the manual approach. The rules for defining these objects need to be respected. They can be found in (AutoCAD documentation, 2019).

Creating linetypes

Although AutoCAD contains more than 80 in-built linetypes, they do not cover the standards in many countries, including Bulgaria. For different purposes, e.g. for the needs of the cadastre, telecommunications, architecture and others, new types of lines need to be created. There are two kinds of linetypes: simple and complex. They are described in text files with the extension lin.

Similar to the work with shapes, in all contemporary versions of AutoCAD, provided Express Tools are installed, linetypes can be created in two ways: manually and automatically, using the command MkLtype. Experiments show that the results of working with the command MkLtype correspond to the results of manually generating linetypes to a sufficient degree. This makes the command comfortable to use

and thus, there is no need to be familiar with the structure of lin-files.

Creating hatch patterns

Two text files are attached to AutoCAD, each of them containing the definitions of more than 60 hatch patterns. Yet, in many cases the creation of new hatch patterns is necessary. In Figure 3 some patterns created for the needs of geology are shown.

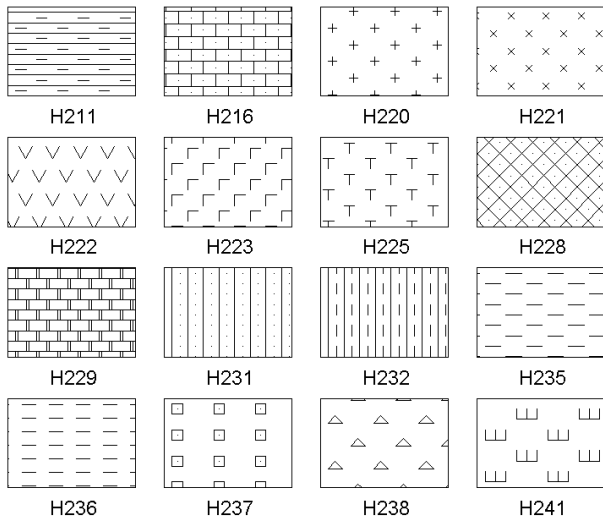


Fig. 3.

Definitions of hatch patterns are written in a text file with the extension pat. Creating such files is a difficult and time consuming process. For example, the definition of the in-built SWAMP hatch pattern in AutoCAD consists of the following text:

```
*SWAMP, Swampy area
0, 0,0, .5,.866025403, .125,-.875
90, .0625,0, .866025403,.5, .0625,-1.669550806
90, .078125,0, .866025403,.5, .05,-1.682050806
90, .046875,0, .866025403,.5, .05,-1.682050806
60, .09375,0, .5,.866025403, .04,-.96
120, .03125,0, .5,.866025403, .04,-.96
```

This is one of the simple patterns, since it contains only 6 lines of description. There are hatch patterns with more than 40 lines in their definitions (e.g. GRAVEL).

SuperHatch command of Express Tools can be handy when hatching. It doesn't generate a file defining the hatch but directly hatches the closed area of the current drawing. This command is especially useful when a closed area is to be hatched with a kind of texture.

When the hatch pattern is an aggregate of linear objects that are evenly/regularly spread within the hatched area especially when they are a checkered pattern then the results provided by SuperHatch command are unsatisfying. On fig. 4a and Fig. 4b the results of hatching using additionally defined hatch pattern (Fig. 4a) and by SuperHatch command (Fig. 4b) are shown.

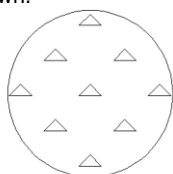


Fig. 4a.

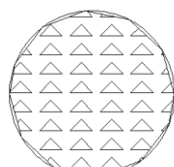


Fig. 4b.

The hatch pattern on Figure 4a stands for galena mineral (triangles with fixed dimensions spread in a checkered pattern). Such a result is impossible to be obtained by SuperHatch command. Furthermore, the SuperHatch command needs the hatching image to be created as a block before the command entering or the image should be uploaded from an external file.

The example above shows that no matter how easy the dialogue with SuperHatch is there are cases when manual creation of hatch patterns is necessary. This is valid especially in the cases when applications for other users have to be developed in AutoCAD. Description of pat-files can be found in (Creating custom hatch patterns, 2019).

Creation of supplementary commands. Programming languages for AutoCAD

The functionality of AutoCAD can be extended using several programme languages. Nowadays, supplementary commands are created mainly using AutoLisp/Visual Lisp, ObjectARX and .NET. Script files, action macros (automatically generated macros), menu macros and code using DIESEL language can hardly be applied to important applications due to the limited abilities of these means. The application of VBA (Visual Basic for Applications) was limited when .NET appeared. VBA was replaced by VB.NET.

AutoLisp and Visual Lisp are useful for creating commands with a simple graphical interface that do not have many arithmetic calculations, but the use of built-in AutoCAD commands or extraction of information from the graphical database prevails (Trifonova et al., 2015). These languages offer extremely powerful and flexible means for choosing graphic objects corresponding to fixed requirements. This is the function ssget (Trifonova, 2015). For example, to generate the selection set of all red circles and all green lines it should be typed in as follows:

```
(ssget "X"
(
(-4 . "<OR")
(-4 . "<AND") (0 . "LINE") (62 . 3) (-4 . "AND>")
(-4 . "<AND") (0 . "CIRCLE") (62 . 1) (-4 . "AND>")
(-4 . "OR>")
)
)
```

Geometric restrictions can be imposed on the selected objects as well, for example, in order to generate the selection set of all lines with starting point having the following coordinates $x=3$, $y>10$ and arbitrary z value the following should be typed in:

```
(ssget "X" '((0 . "LINE") (-4 . "=>,*") (10 3.0 10.0 0.0)))
```

To do the same task using ObjectARX or .NET would be a heavy procedure. It is not by chance that the number of AutoLisp/Visual Lisp files in "Express Tools" is over 78% out of the total number of the files in it.

The main disadvantage of AutoLisp/Visual Lisp in comparison with ObjectARX and .NET are the difficulties related to the creation of dialogue boxes due to the limited abilities of DCL (Dialog Control Language). In addition, ObjectARX and .NET are event-oriented and this type of programming has many advantages.

ObjectARX and .NET are more used nowadays due to the fact that in most universities procedural programming languages like C++ and C# are taught and functional programming languages like AutoLisp/Visual Lisp are rarely discussed.

The optimal option when plug-ins for AutoCAD are developed professionally in AutoCAD is creation of projects where files are generated using different languages so that it can be made use of the advantages of each one of them.

Conclusion

Optimisation of the work of the users of Autodesk products, especially when they work in a team, depends both on the suitable adjustment of the environment assignment and the presence of necessary new elements to be used.

Usage of contemporary programming languages can extend to a large degree the functionality of Autodesk products. A detailed knowledge on programming languages in Autodesk environment allows the process of creation of plug-ins to be optimised.

References

- AutoCAD documentation*; <https://www.autodesk.com>.
Creating custom hatch patterns; <https://allaboutcad.com/creating-custom-hatch-patterns/>.
- Guneva, M., M. Trifonova. 2001. Razshiryavane na funktsionalnite vazmozhnosti na AutoCAD 2000 za avtomatizirano proektirane na kabelni traseta. – *TELEKOM 2001*, Varna, 122–127 (in Bulgarian).
- Top Ways to Customise AutoCAD*; <https://www.autodesk.com/campaigns/inspired-by-autocad/customize>
- Trifonova, M. 2015. *Algoritmi za komputarna grafika i izpolzvaneto im za reshavane na inzhenerni zadachi*. Izd. kushta "Sv. Ivan Rilski", Sofia, 108 p. (in Bulgarian).
- Trifonova, M., E. Nikolova, S. Toncheva. 2015. Development of Applications in Autodesk Products Environment. – *Annual of the University of Mining and Geology "St. Ivan Rilski"*, 58, 4, 99–101 (in Bulgarian with English abstract).