

Comparative Study on Secondary Development Methods of AutoCAD: Performance Comparison and Application of C#, VB, and ObjectARX

Tianxing Chen^{1,*}, Yongchuan Ren², Guofeng Xie², Feng Liu², Kaihuai Luo¹

¹College of Mechanical Engineering, Southwest Jiaotong University, Chengdu, China

²Civil Aviation Logistics Technology Co., Ltd.

*Corresponding author e-mail: txchen@swjtu.cn

Abstract: This paper explores methods and technologies for secondary development using AutoCAD in the field of drafting. By conducting an in-depth study of secondary development methods, the paper aims to enhance software customizability and scalability to meet specific design requirements across different industries. The focus is on three secondary development techniques, including programming languages and interfaces like ObjectARX, VB, and C#.NET. Specific application cases in the field of mechanical drafting are analyzed in detail.

Keywords: AutoCAD, Secondary Development, Scalability, Programming Interfaces.

1. Introduction

As a powerful computer-aided design (CAD) tool, AutoCAD is widely used in mechanical and architectural fields. However, its standard features cannot fully meet the demands of specialized industries[1]. Therefore, secondary development technologies have become a critical means to achieve automation and customized design, improving design efficiency and quality. This paper focuses on three AutoCAD secondary development methods—VB, ObjectARX, and C#.NET—evaluating their performance, compatibility with AutoCAD, and overall development workflow efficiency. By comparing the advantages and disadvantages of these methods, the paper provides developers with guidance for selecting the most suitable solution for specific project needs.

2. Research on the Secondary Development Method of AutoCAD

2.1. The Method of Secondary Development of AutoCAD with VB

The VB software package provides various extension modules called components. These components are created using ActiveX technology and consist of program code and data composed of one or more objects. By utilizing extension

components, VB has gained powerful additional functions, enabling it to operate various application software such as Excel, Word, etc. By using AutoCAD ActiveX components, VB can obtain control handles for operations on AutoCAD, thereby directly controlling a series of operations on AutoCAD[2]. When programming with VB language for secondary development of AutoCAD, the first step is to connect VB with AutoCAD and run it. This requires the following two steps: referencing object libraries in the programming environment before writing code; Write VB program code, create AutoCAD objects, and start running AutoCAD. After connecting VB with AutoCAD, the properties and methods of the object and its subordinate objects can be utilized to achieve graphic drawing and editing operations in the AutoCAD environment using VB language[3].

2.1.1. Compatibility Between Visual Studio and AutoCAD

When writing the secondary development program, it is necessary to ensure that the selected Visual Studio version is compatible with the AutoCAD version, and understand the corresponding AutoCAD API of the programming language used and the available development tools and libraries. Table 1 shows the compatibility between common versions of Visual Studio and AutoCAD:

Table 1. Example of VS Version Compatible with Corresponding AutoCAD Version

Serial Number	Visual Studio Version	AutoCAD Version	Language Type
1	Visual Studio 6.0	AutoCAD2000-2002	Visual Basic 6.0、 Visual C++ 6.0
2	Visual Studio2005-2008	AutoCAD2004-2008	VB.NET,C#
3	Visual Studio2008-2012	AutoCAD2009-2013	VB.NET,C#
4	Visual Studio2012-2019	AutoCAD2014-2023	VB.NET,C#

It should be emphasized that the compatibility of AutoCAD version mainly depends on the API it provides and the programming language it supports. Autodesk usually introduces new APIs and features in new versions, and gradually eliminates old APIs or programming languages.

2.1.2. The Process of Secondary Development of AutoCAD with VB

Development platform: Windows platform Visual Studio2017, AutoCAD2022, The language type is Visual Basic.

Development Environment: Before proceeding with secondary development, it is necessary to ensure that the COM components of AutoCAD 2022 are referenced in the project. Once this is done, you can begin using AutoCAD's objects and functions for secondary development. To add the AutoCAD reference library to a Visual Studio project, you can follow the steps below:

1. Open Visual Studio 2017;
2. Create a new VB project or open an existing project;
3. Right click "Reference" -> "Add Reference" in the project;
4. Find the COM component of AutoCAD 2022 in the pop-up window and add it to the project. Autodesk AutoCAD 2022 Type Library is usually found in the COM tab.

Taking the creation of the "Line" project as an example, Figure 1 illustrates the secondary development process in detail through the flow chart.

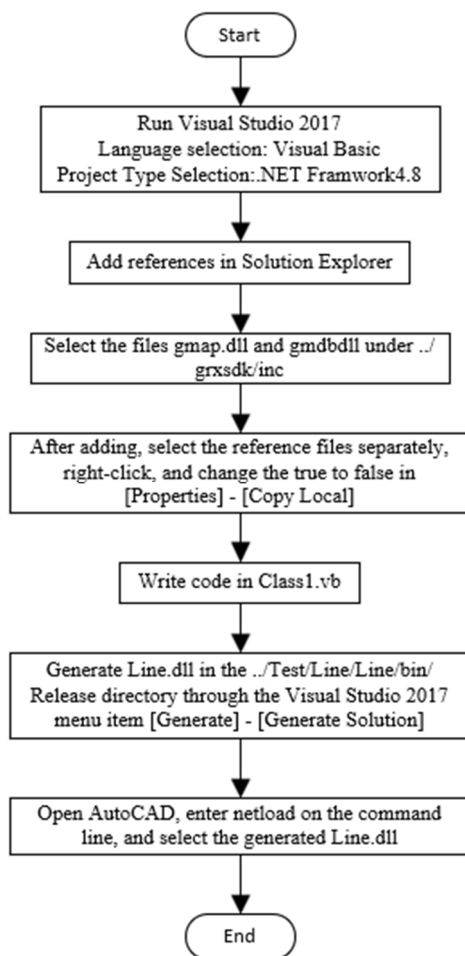


Figure 1. Flow chart of secondary development of AutoCAD using VB

2.2. The Method of AutoCAD Secondary Development with ObjectARX

ObjectARX is an object-oriented C++ application development mechanism, which makes full use of the inheritance, encapsulation and polymorphism of C++ classes. With these features, it significantly improves the security of applications and code reusability. In addition, ObjectARX also supports MFC basic class libraries, enabling developers to efficiently implement many complex functions[4].

AutoCAD's drawing database is an object manager, responsible for managing drawings and their attribute data. Developers can add new management and mechanisms according to application requirements and integrate them into this system. In the integrated development environment, it is a common practice to create a new ObjectARX project [5]. First, select the appropriate project type, such as DLL or application, and then set the basic properties and configuration of the project. This process provides a basic framework for developers to use ObjectARX features to build customized CAD applications.

2.2.1. Basic Structure of Program

ARX applications usually exist in the form of dynamic link libraries. AutoCAD uses `acrEntryPoint()` to call ARX programs. This function is not only the entrance for communication between AutoCAD and ARX, but also serves as the channel for ARX applications to send messages to the system and return status codes to AutoCAD through it. All call requests implemented through the function `ads_defun()` are processed through `acrEntryPoint()`. This design makes `acrEntryPoint()` play a key role in the entire ARX application, responsible for managing and scheduling the interaction and communication between ARX programs and AutoCAD[6].

The messaging process between ARX and AutoCAD is almost completely one-way (from AutoCAD to ARX programs). In the function definition of `acrEntryPoint()`, a switch statement solves the message sent from AutoCAD[7], executes appropriate operations on each message and returns the integer status value.

2.2.2. Compatibility of ObjectARX Version with Corresponding AutoCAD Version

ObjectARX is a powerful toolkit provided by Autodesk for developing AutoCAD plugins and customized applications. The compatibility between ObjectARX versions and AutoCAD versions is crucial. As AutoCAD versions are updated, Autodesk releases new ObjectARX versions that add new features and provide support for the latest AutoCAD features. When developing with ObjectARX, it is recommended to use a version compatible with the target AutoCAD version, for example, using ObjectARX 2023 for secondary development on AutoCAD 2023.

2.2.3. Process of Secondary Development of AutoCAD with ObjectARX

Development platform: Windows platform Visual Studio 2017, ObjectARX2022, AutoCAD2022.

Development environment:

Configure Visual Studio:

1. Ensure that C++ development tools and workloads are selected in Visual Studio;
2. Download and install the development kit (SDK) for ObjectARX 2022;
3. In Visual Studio, set the correct project properties to reference the related library files of ObjectARX 2022 SDK and AutoCAD;
4. Set the project's include directory, library directory, and linker settings to ensure that the ObjectARX library file and the header file provided by AutoCAD are correctly linked.

During development, Visual Studio can be used for debugging and testing, and the developed plug-in can be loaded into AutoCAD 2022 for verification.

Taking the creation of "Line" project as an example, Figure 2 shows the creation process through the flow chart.

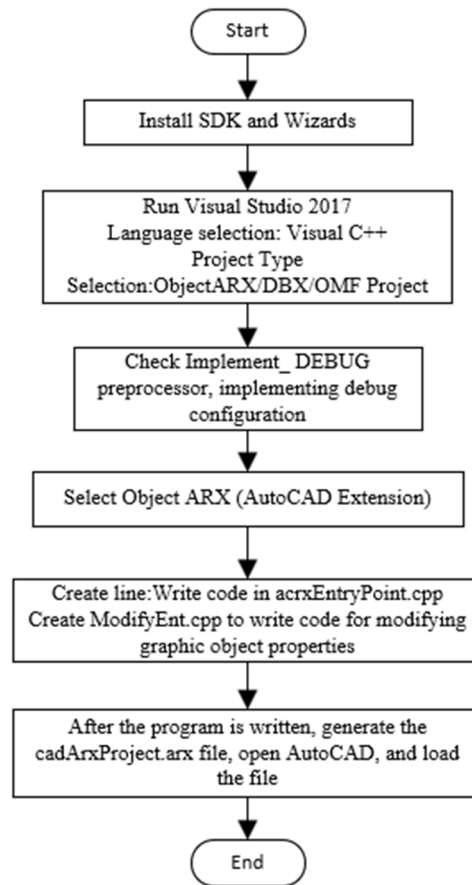


Figure 2. Flow Chart of AutoCAD Secondary Development with ObjectARX

2.3. Methods for Secondary Development of AutoCAD Using C#.NET

C# application and AutoCAD can run in the same process space, and communicate with AutoCAD system in the shared address space by obtaining the instance of AutoCAD application. This method allows applications to send messages to AutoCAD objects by using the message mechanism of Windows, and directly call AutoCAD core functions[8]. With the help of AutoCAD core database structure, graphics system, geometric modeling core and its code, new commands can be established in the same way as the inherent commands of AutoCAD itself, thus improving the operating efficiency and stability.

In C# projects, it is usually necessary to reference the CAD

software NET API library. These libraries are NET assembly, you can access the classes and methods in the API by adding references[9]. Provided by using CAD software NET API, You can write code in a C# project to perform the required operations. These operations may involve creating, modifying, deleting CAD objects, performing data processing, exporting, importing, and other functions. In this way, developers can use the interactivity of C# and CAD software to realize various customized functions and operations.

2.3.1. Compatibility Between C#.NET and AutoCAD

C# is a popular programming language NET framework can be used together to develop customized applications and plug-ins for AutoCAD. Table 4 is the common C#.NET version and the corresponding AutoCAD version.

Table 2. C#.NET version compatible with corresponding AutoCAD version

Serial Number	Visual Studio version	AutoCAD version	.Net Framework version
1	Visual Studio6.0	AutoCAD2004-2008	.Net Framework2.0 and 3.5
2	Visual Studio2005-2008	AutoCAD2009-2013	.Net Framework3.5 and 4.0
3	Visual Studio2008-2012	AutoCAD2014-2019	.Net Framework4.0 and 4.5
4	Visual Studio2012-2019	AutoCAD2020-2023	.Net Framework4.5 and 4.8

It should be pointed out that the compatibility between C# version and AutoCAD version is generally affected by NET Framework support. When selecting the C# version, it is recommended to consider the recommendations in the AutoCAD documentation NET Framework version, and check the corresponding version of NET Framework documentation Learn about the AutoCAD versions it supports.

2.3.2. The Process of Secondary Development of AutoCAD Using C#.NET

Development platform: Windows platform Visual Studio2017, AutoCAD2022, .Net Framework4.8, Language type is C#.

Development Environment:

1. In Visual Studio, by adding a new version of AutoCAD 2022 NET API to access the object model and functions of

AutoCAD;

2. You can search and install AutoCAD 2022 in the NuGet package manager NET API. In the development process, you can use the debugging tools of Visual Studio to test, debug

and verify the interaction between the developed C# plug-in and AutoCAD 2022.

Figure 3 illustrates the process of creating a "Line" project through a flow chart. The specific process is as follows.

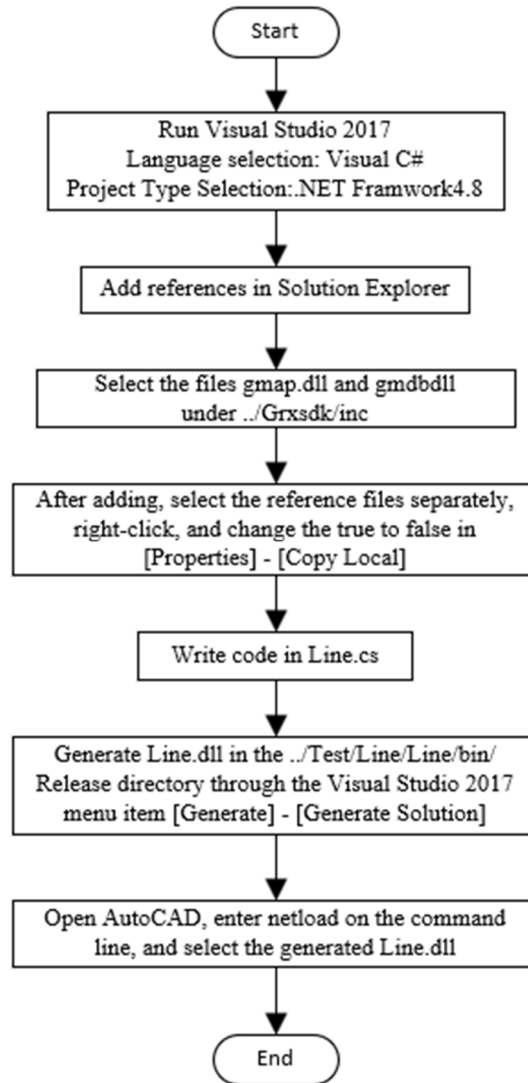


Figure 3. Using C#.NET to AutoCAD secondary development flow chart

2.4. Compare the advantages and disadvantages of secondary development methods

VB, ObjectARX, and C#.NET is a different tool and

technology for AutoCAD secondary development. Each technology has its unique advantages and limitations.

The ability range of three different methods for AutoCAD secondary development and the time required for learning are shown in Figure 4 and Figure 5, and the specific development ability is shown in Table 3.

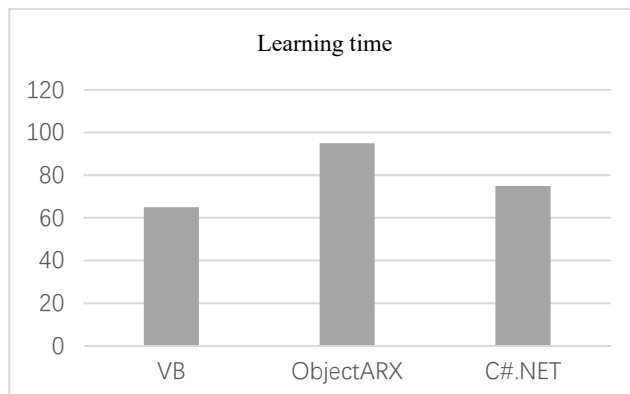


Figure 4. Proportion range of development mode

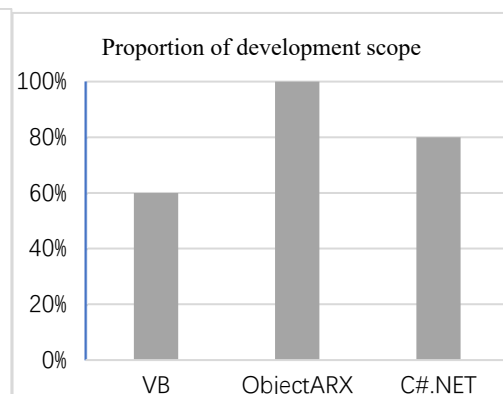


Figure 5. Development Mode Learning Time

Table 3. Capabilities for secondary development of AutoCAD

Secondary development function	VB	ObjectARX	C#.NET
API Operability	Yes	Yes	Yes
Advanced user interface	Yes	Yes	Yes
Underlying events	Limited	Yes	Yes
Advanced API access	Difficult	Yes	Yes
Sophisticated exception handling	Yes	Yes	Yes
64TH PLACE / UNICODE SUPPORT	Yes	Yes	Yes
Garbage collection	Yes	Yes	Yes
Mixed language development	Yes	Difficult	Yes
Hybrid API's interactivity	Yes	Difficult	Yes
The grammar is concise	Yes	No	Yes
Custom entities	Yes	Yes	Yes
Performance	Yes	Yes++	Yes+

Table 4. Compares different tools for the secondary development of AutoCAD

Secondary development tools	Advantages	Disadvantages
Visual Basic (VB)	Visual Basic is an easy-to-learn and easy-to-use programming language, especially for beginners; Simple applications and macros can be created quickly, making them a great tool for rapid prototyping [10]; Seamless integration with other Microsoft Office applications makes it easy to share data across platforms.	Compared to other programming languages, VB has a certain performance penalty; For complex secondary development needs, it will be limited by functions; It can also be limited when dealing with projects with large or high-performance requirements.
ObjectARX	ObjectARX provides full access to AutoCAD's internal APIs, allowing developers to write highly customized applications; Compared to other technologies, ObjectARX can provide higher performance; Allows developers to write highly customized features and tools for specific needs.	ObjectARX requires a lot of time and effort to learn and master; It usually takes longer to develop an application that uses ObjectARX.
C#.NET	C# is a modern programming language with powerful features and excellent performance, and the .NET Framework provides a rich set of libraries and tools to support the development of complex applications. C# has a slower learning curve than other languages, and its syntax is clear and concise.	Installation required. .NET Framework to run developed applications; Compared to low-level languages such as C++, C# has slightly lower performance.

2.5. VB, ObjectARX, and C#.NET to AutoCAD Secondary Development Performance

VB, Object, and C#.NET carried out the secondary development of AutoCAD2022, after creating 1000 connection points (1,1,0) to (1001,1001,0) straight lines through code, the time function was used to obtain the start and end times, and the running time was calculated. The actual running time will vary depending on the computer performance and AutoCAD load, and the efficiency of the three different development methods will be compared to finally choose the best development method.

Use VB to write code to get AutoCAD documents, cycle to generate 1000 straight lines, use the 'Timer' function to get the start and end times, and calculate the running time, and finally display the running time of the generated straight lines through the message box; Write code in C++, measure the code run time through the 'acutimer' function, and finally output the result to the command line; Write code in C# to generate lines and timing via the 'Transaction' class and 'Stopwatch' class in the AutoCAD API. Finally, the runtime is output to the command line via the WriteMessage method of the Editor class. The results obtained after multiple tests are shown in Table 5.

Whether you're using VB, ObjectARX, or C#.NET, they do it all by calling the APIs provided by AutoCAD. The AutoCAD API provides developers with access to the internal functionality of AutoCAD, and there is no significant difference in the efficiency of API calls between different language encapsulations. Complex projects often involve more calculations and operations, resulting in little difference in program run times between the three languages in this case. This is because the running time of the program is mainly affected by the complexity of the AutoCAD API calls and operations, not the language itself. The running time of a program is also affected by the performance of the computer hardware, and if the computer hardware configuration is similar and the execution environment is the same, the running time of the program in different languages will not differ much.

While these factors result in little variation in program runtimes, different languages and platforms have their own advantages for the development of specific complex projects. ObjectARX (C++) is closer to the underlying operation, providing more room for optimization; C#.NET provides modern language features and easier development tools; VB, as a relatively simple language, is more suitable for some simple operations.

Table 5. VB, ObjectARX and C#.NET to AutoCAD secondary development performance

Secondary development tools	The time it takes to generate the segment	Performance
Visual Basic	2.68s	1. Suitable for beginners or those developers who need to create simple tools quickly; Ideal for projects that require seamless integration with Excel, Word, and other Office applications. 2. For handling complex or large-scale projects, VB's performance is limited because it is not as capable of handling large amounts of data or complex algorithms as other languages.
ObjectARX	2.55s	When dealing with complex tasks or large-scale projects, ObjectARX generally has better performance because it is closer to the internal mechanisms of AutoCAD, allowing for more efficient operations and data processing. Ideal for applications that require direct access to and manipulation of AutoCAD's underlying APIs, such as custom entities and advanced graphics operations.
C#	2.49s	1. When dealing with complex tasks or large-scale projects, C# usually performs well. 2. Use. The advantage of the .NET framework, C#'s ability to efficiently handle large amounts of data and more complex algorithms, may be comparable or even better than ObjectARX's performance. Ideal for applications that require the creation of modern user interfaces and interactive features, such as WPF-based graphical user interfaces.

3. Example of Secondary Development of AutoCAD Using C#.NET

3.1. Development Background

Civil Aviation Chengdu Logistics Technology Co., Ltd. currently produces and manufactures air logistics transportation equipment for customized design products, need to determine the specifications and dimensions of the product according to customer use scenarios, so in the same project the same type of air logistics transportation equipment will exist a variety of specifications, and these specifications of different products except for the specific non-standard design part, the rest of the structural dimensions are no different, even so, in order to clarify the specifications and dimensions of each product, still need to design a large number of drawings, the workload is particularly huge. In order to improve the efficiency of drawing and reduce human errors in the process of drawing, the non-standard size change rules of air logistics transportation equipment are sorted out and analyzed, and the CAD drawing software is redeveloped to realize the intelligent generation of drawings of non-standard transportation equipment.

3.2. Overall System Design

The design of this system is based on the traditional linear conveyor design combined with CAD secondary development technology, which converts the operation that originally required manual modification into a non-standard part drawing automatically generated by the system according to the design rules by inputting design parameters. Through the analysis of the traditional design features, the design process of the linear conveyor can be developed by CAD, including the adjustment of the size of non-standard parts, the deployment of bed stiffeners, nuts, round holes and other parts, as well as the marking of each size. In the program, the rules for the non-standard design of the linear conveyor are formulated, so that the system can generate non-standard parts drawings according to the input of non-standard parts parameters and in accordance with the pre-established rules. This will greatly improve the design efficiency of designers and reduce the error rate in non-standard design.

3.3. System Usage

When the system is loaded into the CAD software, after entering the command "Series" in the command line input box, the series selection interface will pop up, as shown in Figure 6, select the series to be designed, and click the cancel button to close the interface.

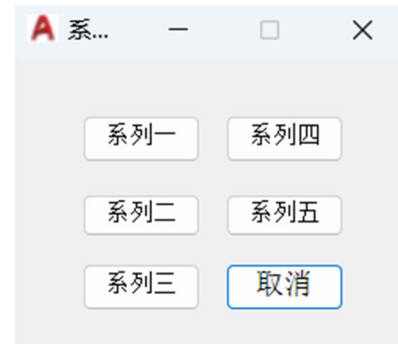


Figure 6. Series selection screen

Select a series on the series selection page or enter design_1200 on the command line to pop up the design interface, as shown in Figure 7.



Figure 7. Parameter input interface for Series 1

In the parameter input box, enter the parameters to be set, and any non-standard part parameter can be input according to the design requirements, and the rest of the non-standard

design part parameters will also change accordingly, and the drawing generation can be completed, as shown in Figure 8.

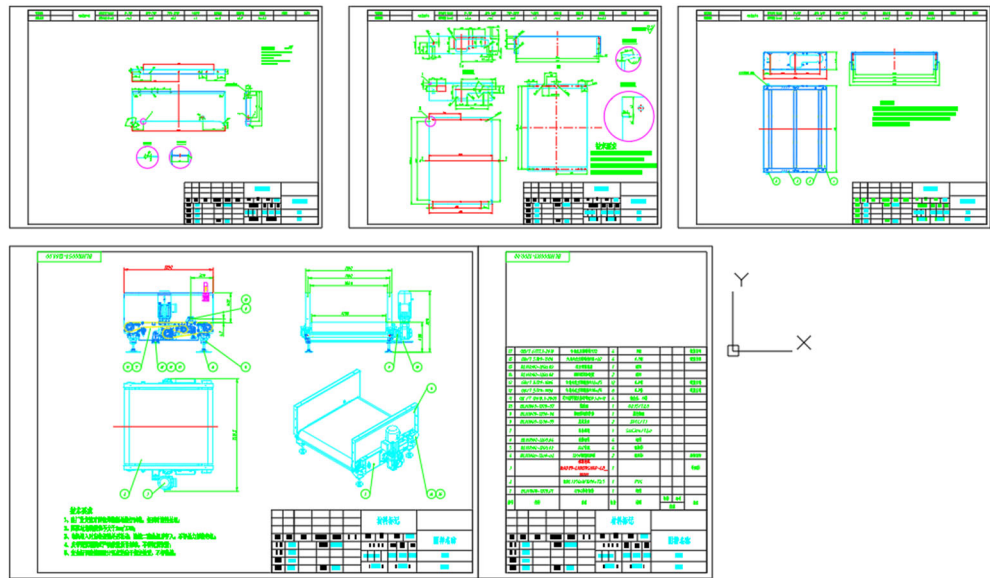


Figure 8. Generating a non-standard part drawing

4. Conclusions and Outlook

C#.NET is more responsive to the secondary development of AutoCAD, and C# provides an easy-to-use multi-threaded programming model that helps to achieve parallel processing and improve the performance and responsiveness of the program, which can be more efficient when dealing with large-scale or complex tasks. C# also easily integrates with other Microsoft technologies, including Windows Presentation Foundation (WPF), Windows Forms, ASP.NET, and more. This allows developers to create richer, more advanced user interfaces and applications. Therefore, this article uses C#.NET carries out the secondary development of AutoCAD related projects.

AutoCAD secondary development provides a wealth of programming tools and interfaces, so that users can expand and customize AutoCAD functions according to their own needs, so as to improve work efficiency and design quality. Secondary development technology is widely used in automated annotation, customized commands, data interaction and processing, etc., providing engineers with more flexibility and convenience. With the continuous development of technology, we can expect to see more secondary development methods based on new technologies such as cloud computing and artificial intelligence in the field of software engineering, which will further improve the customization and extensibility of AutoCAD, and enable the creation of customized solutions based on specific needs and workflows, thereby improving drawing efficiency and reducing human error.

References

[1] Yongchuan R, Guofeng X, Feng L, Kaihuai L, Tianxing C. Research on Secondary Development Method Based on Haochen CAD [J]. Frontiers in Computing and Intelligent Systems, 2024,6(3):41-44.

[2] Fufu Z, Yongchaun R, Feng L, Guofeng X, Tianxing C. Research on the Aided Design System of Linear Conveyor Equipment based on Haochen CAD[J].Frontiers in Computing and Intelligent Systems,2023,5(3):117-120.

[3] Xiaoyan C. Research on the application of CAD technology in mechanical drawing[J].China Equipment Engineering, 2023 (17): 241-244.)

[4] Hongliang W, Xiuyun Z, Lin ZHAN. Batch Drawing Generation by Integrating VB with AutoCAD, 2019, 53(01): 112-116.

[5] Jiaxin C, Qiao H. Automatic drawing system of motor model based on VB for AutoCAD secondary development [J]. Mechatronics Engineering Technology, 2021, 50(11):17-20.

[6] Y. Dai, "Application Research of Interior Design under Computer 3D Simulation Graphics CAD Modeling System," 2023 International Conference on Electronics and Devices, Computational Science (ICEDCS), Marseille, France, 2023, pp. 410-414, doi: 10.1109/ICEDCS60513.2023.00080.

[7] Lamei F. Research on design method of end-of-mine boundary diagram of open-pit mine based on AutoCAD secondary development [D]. Xi'an University of Architecture and Technology, 2023.DOI:10.27393/d.cnki.gxazu.2022.001029.

[8] Chuanjun W. Secondary development of AutoCAD dimension testing system based on ObjectARX[J].Digital Technology and Application,2013,(03):195-196.DOI:10.19695/j.cnki.cn12-1369.2013.03.136.

[9] Li Hong and Chun-Sheng Ma, "Development of common parts and standard parts database based on AutoCAD," 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, China, 2010, pp. V13-467-V13-470, doi: 10.1109/ICCASM.2010.5622855.

[10] Xiaoguang L. Implementation Technologies for Secondary Development Platforms of CAD Software[J].Electronic Technology and Software Engineering,2016(05):59-60.