

Automatic part creation in SolidWorks using API functions

Valentina Dukić¹ [ORCID 0000-0001-7733-1236], Andrea Gutai¹ [ORCID 0000-0002-3315-4108], Andraš Anderla¹ [ORCID 0000-0003-4642-9382], Dušanka Dakić¹ [ORCID 0000-0002-1707-7616], Marko Arsenović¹ [ORCID 0000-0002-1405-7855]

¹Faculty of Technical Sciences, Department of Industrial Systems and Management, Novi Sad

Abstract: *The SolidWorks API (Application Programming Interface) is a set of tools, functions, and interfaces that allows interaction and automation of various aspects in SolidWorks software. By leveraging the SolidWorks API it is possible to automate repetitive design tasks, integrate SolidWorks with other software systems, create custom design applications, extract data for analysis and reporting, and enhance productivity for SolidWorks users. Using macro API calls are recorded to allow multiple executions with a simplified part creation process. This paper presents the use of SolidWorks API and macros for generating part within the SolidWorks environment. With macros is possible to generate a part by providing only input dimension values. After creating the part with the use of the macro recorder, the code was edited and the code for entering the input parameters was defined.*

Key words: SolidWorks, API, automation, part

1. INTRODUCTION

Most of design tasks are routine in nature. The demand for CAD-based (Computer Aided Design) tools and software that can automate common daily tasks has surged as a result. These activities either focus on tiny, straightforward assemblies or bigger, more intricately constructed ones. The primary goal in any situation is to provide high-quality items as quickly as possible, while also saving time. The SolidWorks API can be used to automate the development of parts in SolidWorks, which can greatly increase productivity and simplify design procedures. Applications can be created as well as parts with SolidWorks API, which eliminates the need for manual modeling and repetitive operations. Modern design has made life much easier by reducing time wastage and maintaining proper usage of resources. By using SolidWorks products having the same mechanical characteristics and shapes, the corresponding entity model library can be created through parametric design. This can improve design efficiency and simplify design work (Avitus Titus and Liu Xue Bin, 2013). The SolidWorks API provides a powerful set of tools and functions that enables a user to interact with SolidWorks programmatically. API is a software development tool which enables integration between different applications by providing a code in a programming language within another application (Prince et al., 2003). The Microsoft Component Object Model (COM) technology, which enables cross-language program communication, serves as the foundation for the SolidWorks API. The API provides access to a broad variety of functionality, enabling programmers to automate processes including creating and changing part models, assemblies, and drawings as well as accessing and editing features, dimensions, materials, and other design elements. API is a tool to write a code in a programming language within another application. As a result, a direct integration between different applications can be developed (Prince et al., 2003). SolidWorks is one of the applications which support API with different programming languages such as C++, Visual Basic, and Visual Studio. SolidWorks API automates the design and the assembly operations by creating codes in a specific programming language and it has been applied for different design methods (Farhan et al., 2012). The following are typical tasks that can be completed utilizing the SolidWorks API: generating new drawings, parts, or assembly documents, adjusting the dimensions, adding features, or modifying the characteristics of an existing document, creating design variations or carrying out design assessments using automation, obtaining data from SolidWorks documents, such as dimensions, mass characteristics, or a bill of materials, to improve the SolidWorks user experience, bespoke user interfaces and forms are created and interacting with databases, other software programs, or external systems. This article discusses the benefits of using the CAD system's programming interface to build custom programs (macros) that can help users to automate a variety of processes using a straightforward graphical user interface (GUI). This shortens the design process and obviates any potential errors. There are many CAD software packages with API functionality. SolidWorks CAD is chosen in this paper due to its user-friendly features and support for API capability. The paper is organized as follows: related work is described in Section 2. Methodology for part automation using SolidWorks macro

API is presented in Section 3., Section 4. Represent the results of work. The research is briefly reviewed in the final section, which also offers closing thoughts and suggestions for future work.

2. RELATED WORK

Numerous academics have used the API of numerous CAD systems to create tools and applications, and they consistently came to the conclusion that the benefits and advantages of the API are substantial.

A bibliometric analysis technique is presented by Ciric D., Lalic B. et al. (Ciric et al., 2021) to review mass customized scientific production. The most successful journals are mostly those devoted to industrial engineering, production, operational, and manufacturing research; among the ten most significant publications are those that are frequently printed in management sciences-focused journals. The following list of pertinent publications comes from the subject area covered in the aforementioned study. Esanakula Jayakiran Reddy et al. (Reddy, E. J. et al., 2016) researched knowledge based intelligent design system for bearings library construction using SolidWorks API. Bearing's dimensions change during the design process depending on the working or loading conditions anytime an assembly component is changed, but their geometry does not. It takes time and effort to recalculate the appropriate dimensions each time a component is changed. A computer program is created for design calculation purposes in order to reduce time consumption. To create the CAD model, the output of this program is connected to the SolidWorks program via a database. Design and modeling tasks have been automated as a result. The appropriate bearing is chosen once the user has entered the input data and clicked the Calculate button to calculate it based on the loading conditions in the background using created computer programs and intelligent design systems as well as standard imperial formulas. The GUI's combo box field shows the outcome. After the Model button is clicked, the CAD model is generated.

With this work, Yayakirin tried to develop an intelligent design system and succeeded in reducing the modeling time. However, the presented example is simple and applicable to simpler bearings. An additional problem may be the software version. The SolidWorks API is tied to certain versions of the software, and if the software is updated, it is necessary to adapt the code.

K.Balachandar et al. (K.Balachandar, 2020) show macro assisted design automation of piston and connecting rod using SolidWorks API. They defined the design automation steps shown in Figure 1.

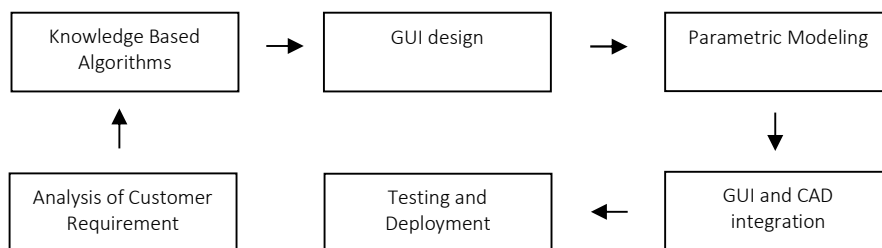


Figure 1: Design Automation Steps

Systemic risks in the industry of engineering are demonstrated by Djakovic M. et al. (Dakovic M. et al., 2020). The paper's main points are delays and time spent on each activity.

Abstraction levels and corresponding design levels are summarized in 6 basic steps by Sadiq M. Sait and Habib Youssef (Sait and Youssef, 2006). CAD subproblem level is idea, problems at this level are solved with CAD tools. Followed by architectural level based on architectural design. The third step is on logic level and represents logical design, next is physical design and finally fabrication.

Ciric D. et al. (Danijela Ciric et al, 2020) researched on the use of ICT solutions in Serbian industrial businesses. The study revealed that implementation of all or at least some parts of Industry 0.4, including the use of various software solutions, is required for Serbian businesses to begin the process of reindustrialization and keep up with trends.

Amel Meddahi et al. (Amel Meddahi et al., 2009) explained SolidWorks API offers several functionalities to define the suitable trajectory of the robot. The trajectory is related with the relative position and orientation of the end-effector and 3D encountered object model.

Fang Zhou and Zhang Li-Na (Fang Zhou and Zhang Li-Na, 2012) defined a transformation from a VB script to a VC++ program on the caliper. The first step of modeling based on the SolidWorks API is to record the modeling process using the SolidWorks macro recording tool followed by transformation into a VC++

program. After creating the model and recording the modeling process from the Macro recorder, since the recorded codes are VBA codes, they are transformed into VC++ codes for further development.

Although the SolidWorks API is used in all areas, it is essential to take a look back at earlier experiences, methodology, and technologies in order to draw meaningful conclusions from ongoing research and to advance this subject. In the following work, the application of macros is shown, the VBA (Visual Basic for Applications) code is explained and cleaned.

3. METODOLOGY

Before starting the creation of the model, recording with the help of macros was started. To record and replicate actions taken with the SolidWorks interface, the SolidWorks macro is used. A macro contains calls to the API that are equivalent to operations performed in the user interface. The macro enabled the recording of mouse movements, menu decisions, keystrokes, and zooming in and out of the model. Macro commands are accessed via Tools where the Macro Menu is located. Selecting Macro recording in the upper left corner opens the commands for starting, stopping and stopping the recording. Before starting the creation of the model, the recording was started by clicking on Run. Appearance of running macros is shown in Figure 2.



Figure 2: Macro recording

For the purposes of this work, a model was created that represents the flashlight shown in Figure 3.

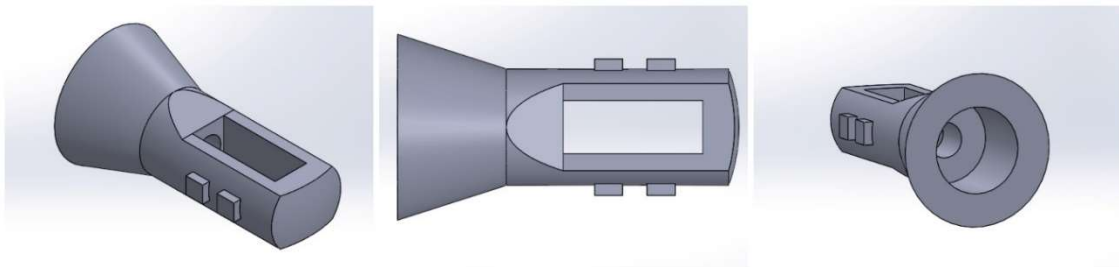


Figure 3: Example – flashlight

After the model is drawn the macro is saved. From the macro menu, Run is selected and the previously saved macro is selected. This starts the process of making the model according to the previously completed steps. All the steps that needed to be done to draw the flashlight are automated in two steps (Play and selecting the required macro).

3.1 API methods used in methodology

The code is explained from start to finish below. A more detailed examination of the macro code shows how powerful and easy to understand macro codes are. Macro code usually needs to be edited to complete automation task. The following text explains the lines of code that show key steps of creating a flashlight model. There are two procedures in the code, declarative and main. In declarative procedures, input lines of code that are necessary are displayed:

Dim swApp As Object

Dim Part As Object

Dim boolstatus As Boolean

Dim swApp As Object - This line declares a variable named swApp of type Object. The swApp variable is typically used to represent an instance of the SolidWorks application.

Dim Part As Object: This line declares a variable named Part of type Object. The Part variable is typically used to represent a SolidWorks part document.

Dim boolstatus As Boolean: This line declares a variable named boolstatus of type Boolean. The boolstatus variable is typically used to store a true/false value indicating the status of a particular condition.

In SolidWorks automation, these lines of code are often used to declare and initialize variables that will be used in later code logic. While Boolean is used to store logical values, the Object data type is flexible enough to store a variety of object kinds.

The next block represents the connection to SolidWorks, necessary for the model to be clickable:

Sub main()

Set swApp = Application.SldWorks

Set Part = swApp.ActiveDoc

Set swApp = Application.SldWorks: This line assigns the swApp variable the reference to the SolidWorks application. Application represents the current instance of the SolidWorks application, and SldWorks is a property or method that returns the instance of SolidWorks. By setting swApp to Application.SldWorks, the swApp variable is initialized to hold the reference to the SolidWorks application.

Set Part = swApp.ActiveDoc: This line assigns the Part variable the reference to the active document in SolidWorks. swApp.ActiveDoc returns the currently active document in SolidWorks (which can be a part, assembly, or drawing). By setting Part to swApp.ActiveDoc, the Part variable is initialized to hold the reference to the active document. These lines of code are often used at the beginning of a SolidWorks VBA macro to set up the necessary references to the SolidWorks application and the active document.

In the following bit line of code:

boolstatus = Part.Extension.SelectByID("Line1", "SKETCHSEGMENT", 9.06096124160229E-02, 2.1781156830776E-04, 0, False, 0, Nothing, 0)

The SelectByID2 method is used to locate a sketch segment with the name "Line1" at the supplied coordinates in the sketch plane. The outcome is saved in the boolstatus variable. Next block of code "Line1" is the name or identification of the chosen object, next "SKETCHSEGMENT" is the kind of object that needs to be chosen. Finally, the coordinates for X, Y and Z are defined and extra parameters depending on the situation.

Next important line: **Set skSegment = Part.SketchManager.CreateLine(0#, 0#, 0#, 0#, 0.039624, 0#)**

First *Set skSegment* shows that the resulting line sketch segment object will be stored in the variable skSegment, which is declared and initialized in this line.

Part.SketchManager alludes to the Part object's corresponding sketch manager object. A part or component is represented by the Part object in CAD (Computer-Aided Design) software.

CreateLine() to create a line sketch segment, execute the createLineSketch method on the SketchManager object. If some other tool was used, such as 3PointArc, SelectedByID, the name of the used tool would be written in place of CreateLine. The rest of the line of code remains the same (the parameters for X, Y and Z change). And (0#, 0#, 0#, 0#, 0.039624, 0#) represent the coordinates of the line segment.

The creation of the revolve feature is shown in the following line of code:

Set myFeature = Part.FeatureManager.FeatureRevolve2(True, True, False, False, False, False, 0, 0, 6.2831853071796, 0, False, False, 0.00254, 0.00254, 0, 0, 0, True, True, True)

Set myFeature: This variable is declared and initialized in this line to hold the finished revolve feature object.

Part.FeatureManager: It alludes to the part object's connected feature management object.

FeatureRevolve2: This function creates a revolve feature for the FeatureManager object.

The next important part of the model is the opening inside the flashlight handle. Openings, holes and cutouts in SolidWorks are created using the Extrude Cut tool. The following line of code shows the application of the mentioned tool.

Set myFeature = Part.FeatureManager.FeatureCut4(True, False, False, 1, 0, 0.00254, 0.00254, False, False, False, False, 1.74532925199433E-02, 1.74532925199433E-02, False, False, False, False, True, True, True, True, False, 0, 0, False, False)

Within this line, the arguments that are passed to the Cut Extrude method are defined. The parameters and settings for the FeatureCut4 operation within the CAD software are defined using these arguments. The software and its associated documentation will determine the precise significance and effect of each argument. The opposite operation to Cut Extrude is Boss Extrude. Boss Extrude feature is used to create a solid feature by extruding a sketch in a specific direction. In code, the Boss Extrude operation is formulated in the same way as Extrude Cut and is passed the same parameter types. Another key operation used in the creation of the model is Mirror, which was used on the finished model. A feature or collection of features can be duplicated over a chosen plane or planar face using the Mirror feature.

Set myFeature = Part.FeatureManager.InsertMirrorFeature(False, False, False, False)

In this line, mirror features are created in the active document based on the selected features.

4. RESULTS

The specific advantages of automated component generation in SolidWorks employing API capability are highlighted in detail by the findings. The study initiative raised user happiness, enhanced design

effectiveness and quality, and enabled breakthroughs in CAD automation inside the company. Main result of presented research is reduction in design time. Reducing design time was the main goal of automated part creation. The research produced a considerable decrease in design time across diverse test cases by using API-driven automation scripts. The automated procedure was, on average, 30% quicker than the manual part fabrication. For difficult or repeated design projects, this time savings is especially substantial. Human mistake in part creation was removed by automation. Dimensions, clearances, and material requirements were all applied consistently in line with predetermined criteria. Engineers that used the automated technique noticed a significant increase in the uniformity of the designs across several projects. Errors that result from manual part fabrication are frequently expensive and time-consuming to fix. Automation, on the other hand, greatly decreased the frequency of design faults. Fewer design errors mean fewer prototyping and manufacturing errors, resulting in reduced production costs.

5. CONCLUSIONS

The integration of systems and applications as well as the automation of repetitive tasks are all made possible by the SolidWorks API. Because API makes it possible to expedite design processes, reduce error rates, and work more effectively, productivity has grown. Customized plug-ins and macros can automate time-consuming tasks, create complex geometry, and execute simulations or analysis. With macro recording, automating repetitive tasks is easy to do without any programming skill. Despite the advantages of both macro recording and the API, the SolidWorks API offers greater customization, flexibility, and control over the design process. Users can increase productivity by automating tasks, create workflows that are specific to their design processes, improve accuracy and consistency in design, access advanced design features outside the scope of the default user interface, and integrate SolidWorks and other software systems. Users are able to work more productively, handle challenging design problems, and produce outputs of greater quality because to these advantages. Results:

REFERENCES

Avitus Titus, Liu Xue Bin, Secondary Development of Solid works for Standard Components Based on Database. International Journal of Science and Research (IJSR).

Balachandar, K., Gangadharan, J., Arjun, S., Jayaraman, P., Krishna, V. E., Joshuva, A., & Aravinth, S. (2020). Macro Assisted Design Automation of Piston & Connecting Rod Using Solidworks API. International Journal of Mechanical Engineering, 83, 12964-12969. Published by The Mattingley Publishing Co., Inc.

Ciric D., Lalic B., Marjanovic U., Savkovic M., Rakic S. (2021). A Bibliometric Analysis Approach to Review Mass Customization Scientific Production. In: Dolgui A., Bernard A., Lemoine D., von Cieminski G., Romero D. (eds) Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems. APMS 2021. IFIP Advances in Information and Communication Technology, vol. 634, pp. 328-338, Springer, Cham. ISBN: 978-3-030-85913-8.

Ciric D., Lolic, T., Gracanin, D., Stefanovic, D., Lalic, B. (2020). The Application of ICT Solutions in Manufacturing Companies in Serbia. In: Lalic, B., Majstorovic, V., Marjanovic, U., von Cieminski, G., Romero, D. (eds) Advances in Production Management Systems. Towards Smart and Digital Manufacturing. APMS 2020. IFIP Advances in Information and Communication Technology, vol 592. Springer, Cham. https://doi.org/10.1007/978-3-030-57997-5_15.

Djakovic M., Lalic B., Delic M., Tasic N., Ciric D.: Systematic mitigation model sensitivity in the initiation phase of energy projects. Advances in Production Engineering and Management, 2020, vol. 15, no. 2, pp. 217-232. ISSN 1854-6250. (Engineering, Manufacturing; 23/50); IF 2020 = 3,382 <https://doi.org/10.14743/apem2020.2.360>.

Fang, Z., & Zhang, L. N. (2012). Development of Standard Part Library Based on SolidWorks. *Advanced Materials Research*, 549, 1041–1045. <https://doi.org/10.4028/www.scientific.net/amr.549.1041>

Farhan, U. H., O'Brien, S., & Tolouei Rad, M. (2012). SolidWorks Secondary Development with Visual Basic 6 for an Automated Modular Fixture Assembly Approach. International Journal of Engineering, 6(6), 290-304. Retrieved from <https://ro.ecu.edu.au/ecuworks2012/680>.

Meddahi, A., Baizid, K., Yousnadj, A., & Iqbal, J. (2009). API Based Graphical Simulation of Robotized Sites. In Proceedings of the 14th IASTED International Robotics and Applications Conference (pp. 1-6). Cambridge, MA, USA: Université de Boumerdes, Algérie; University of Genova, Italy; Italian Institute of Technology (IIT), Genova, Italy.

Prince, S.P., Ryan, R.G., Mincer, T.: Common API: using visual basic to communicate between engineering design and analytical software tools. In: ASEE Annual Conference (2005).

Reddy, E. J., Sridhar, C. N. V., & Rangadu, V. P. (2016). Research and Development of Knowledge Based Intelligent Design System for Bearings Library Construction Using SolidWorks API. In V. P. Rangadu (Ed.), Proceedings of the International Conference on Artificial Intelligence and Evolutionary Computations in Engineering Systems (ICAIECES 2016), Springer Proceedings in Mathematics & Statistics (Vol. 197, pp. 163-172). Springer International Publishing.

Sadiq M. Sait and Habib Youssef (2006) VLSI Physical Design Automation: Theory and Practice. Singapore, World Scientific Publishing.