Technologia i Automatyzacja Montażu

Volume 127, Issue 1/2025, Pages 74-82
https://doi.org/10.7862/tiam.2025.1.6

Original Research

ARTIFICIAL INTELLIGENCE-BASED DESIGN OF ASSEMBLIES IN THE FREECAD SOFTWARE

PROJEKTOWANIE ZŁOŻEŃ Z ZASTOSOWANIEM SZTUCZNEJ INTELIGENCJI W PROGRAMIE FREECAD

Roman WDOWIK [1,*], **Artur BEŁZO** [1]

[1] Department of Manufacturing Processes and Production Engineering, Rzeszów University of Technology, al. Powstańców Warszawy 12, Rzeszów, Poland

* Corresponding author: rwdowik@prz.edu.pl

Abstract

The article presents application of the FreeCAD software and generative artificial intelligence (AI) in the design process regarding exemplary mechanical engineering-related assemblies. Authors have studied how to support a product development and computer-aided (CAD) design phase by the use of generative pre-trained transformer (GPT). Prompting process, that lead to Python language-based code creation in the ChatGPT™ by OpenAI company, was studied from the perspective of 3D assemblies creation in the FreeCAD software. Moreover, authors studied how to improve the basic 3D models by the use of text-based prompts and CAD user improvements of the solid models and discussed the general effectiveness of this approach. It was also separately studied how to automate creation of many parts of the same type by the use of the AI, and apply such CAD libraries in the developed assemblies. The performed research presented selected possibilities of AI-based design process, challenges of the design process and future areas of investigation of AI-based assembly development resulting from the usage of existing GPTs.

Keywords: artificial intelligence, assembly development, fixtures, manufacturing, CAD

Streszczenie

Artykuł przedstawia zastosowanie programu FreeCAD oraz generatywnej sztucznej inteligencji (SI) w procesie opracowywania przykładowych złożeń. Autorzy badali jak wspierać etap projektowania części w zakresie modelowania wspomaganego komputerowo (CAD), wykorzystując do tego celu generatywny wstępnie wyuczony model językowy (GPT). Zbadano proces tworzenia promptów w usłudze ChatGPT™ firmy OpenAI prowadzący do wygenerowania w tym środowisku kodu w języku Python dla potrzeb generowania złożeń 3D w programie FreeCAD. Analizowano także jak ulepszyć bazowe modele 3D przez zastosowanie dodatkowych promptów oraz bezpośrednich zmian wykonanych przez użytkownika z zastosowaniem istniejących narzędzi CAD oraz ogólną efektywność takiego podejścia. Omówiono także oddzielnie jak zautomatyzować opracowywanie wielu części tego samego typu przez zastosowanie sztucznej inteligencji oraz jak wykorzystać takie biblioteki CAD w rozwijanych złożeniach. Wykonane prace uwidoczniły wybrane możliwości modelowania wspomaganego sztuczną inteligencją, wyzwania tego procesu i przyszłe kierunki badań dotyczących projektowania złożeń przez zastosowanie istniejących modeli językowych sztucznej inteligencji.

Słowa kluczowe: sztuczna inteligencja, projektowanie złożeń, uchwyty, wytwarzanie, CAD

1. Introduction

The design process of various fixtures such as chucks, vises, clamps, etc. plays a crucial role in modern manufacturing environments. The fixtures are designed to secure a machined part during the manufacturing process that generates forces which must be transferred by the fixture in order to protect a part from unnecessary displacements. The above-mentioned design process requires designer's skills, appropriate design tools and technological knowledge. Nowadays, the most of the design activities are performed by the use of the 3D CAD software (three-dimensional computer-aided design software) tools which are available for manufacturing firms. The developments in the area of software have changed the design techniques a lot. For instance drawing boards, being utilized in the past, have been replaced by the abovementioned CAD software. Most of them are commercial tools licensed for the final users but also some open-access (free of charge) tools are available. The fixture design requires an analysis of factors linked to the quality of clamped and manufactured parts, cost effective fixture manufacturing process planning and fixture designing and other crucial factors influencing the effectiveness of the entire manufacturing process. In this context, authors selected an open-access software such as FreeCAD stable version 1 (FreeCAD, 2024) to discuss the artificial intelligence (AI) selected applications in the product development process. Accessibility of the Python console in the FreeCAD software and the great progress regarding that software capabilities convinced authors to focus on the presented study.

Nowadays, AI and AI-based tools are becoming very popular among scientists and they promise a game changing perspective in the area of production challenges such as Industry 4.0 challenges (Sarker, 2024). Large Language Models (LLMs) using generative pre-trained transformer (GPT) such as ChatGPT™ by OpenAI (ChatGPT, 2024) or Copilot™ by Microsoft (Copilot, 2024) or Gemini™ by Google (Gemini, 2024) enable to redefine selected design approaches and develop methodologies improving the overall design process in CAD environments. The FreeCAD software offers the abovementioned tool available for a user that enables to create a Python language-based code which can be directly used for the solid models design needs. The tool implemented in the software is a Python console activated from the main menu: *View\Panels*. If that tool is combined with existing generative artificial intelligence-based tools such as text chats enabling to create Python code, the design process has a potential to be aided, however, the methodology of such design process, difficulties linked to it have not been studied so far.

In this context, authors decided to study selected areas linked to the AI-based assembly creation in the FreeCAD software that may contribute to design process improvements and gaining new knowledge in that area.

Currently literature analysis linked to that approach is limited as it was mentioned above, however, several authors tried to initiate the academic discussion in that area. For instance, Badagabettu et al. in their work (Badagabettu et al., 2024) presented Query2CAD, a novel architecture that returns a CAD model similar to the user's prompt. Deng et al. (Deng et al., 2024) from Cranfield University tested LLM for creation of CAD models (simple assemblies of gears) in OpenSCAD software. Hunde and Woldeyohannes (Hunde and Woldeyohannes, 2024) discussed perspectives of implementation AI into CAD environments. Moreover, Kapsalis (Kapsalis, 2024) demonstrated Natural Language Processing for 3D Modelling to Enhance Computer-Aided Design Workflows. In addition, Li and co-authors (Li et al., 2024) presented Multi-Modal Large Language Models for 3D Computer-Aided Design Generation.

In this context, the proposed topic of the paper is an actual research problem in authors' opinion and methodology of AI-based fixture design and analysis of exemplary results shall be indicated as main areas of the study.

2. Methodology of AI-based fixture design in the FreeCAD

This chapter presents the methodology behind the 3D assembly design by the use of the ChatGPT™ and the FreeCAD software having implemented the Python console. Moreover it presents and discusses the basic methodology on how to implement the changes required to adjust the basic 3D models to the design and technological requirements.

The AI-based modelling is supported by the text prompting, Python language-based code creation by the ChatGPT™ and its final running in the FreeCAD software that starts the adjustment processes using available CAD tools. The chapter is divided into a design testing phase and a design improvement phase.

2.1. Testing phase

The first attempts concerned creation of simple prompts in order to verify how detailed are existing knowledge and capabilities of the text LLM about the typical construction details of selected assemblies. This phase is important in authors opinion because it reveals the general problems related to CAD modelling automation by the use of AI.

In this context, only a general request was send to the chat to generate simple assemblies such as a three-jaw chuck having application on turning machines and also for clamping axisymmetric parts on other types of machine tools, a turning tool holder that is applied to clamp a turning insert that cuts material, screw-nut-wrench assembly as a typical assembly, assembly of two gears and vise (Figure 1).

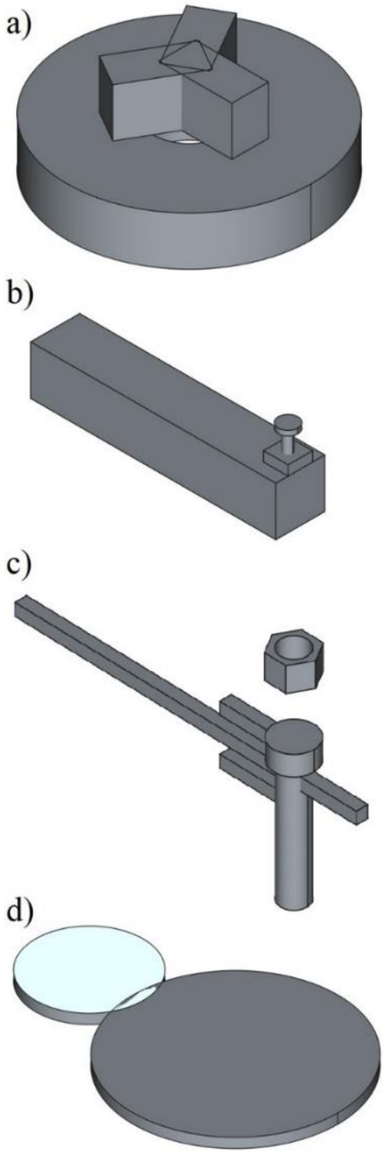

Fig. 1. Selected assemblies created on the basis of simple prompts and the ChatGPT^{TM} text responses – print screens captured in the FreeCAD environment: a) 3-jaw chuck, b) turning tool holder, c) wrench-screw-nut, d) pair of gears

For example, an exemplary simple prompt regarding tool holder was created and run in the ChatGPT^{TM} from the developed text (prompts were tested in Polish language):

Create a lathe tool consisting of a holder, an insert and a mounting screw.

This text request of authors (and similar requests regarding other assemblies) triggered the LLM and generated a Python language-based code.

In some cases (e.g. in the case of screw-nut-wrench assembly) authors requested some small changes such as an additional request (reprompting) regarding improvement to the hexagonal shape of the nut.

The results were recorded as **.FCstd* files – default format of the FreeCAD software. The abovementioned exemplary CAD 3D models generated in the FreeCAD software are presented here without any further changes as print screens saved in the working area of the FreeCAD environment.

It is clearly visible in the Figure 1 that only a few similarities can be indicated if the obtained visual result of the abovementioned parts is compared to the real objects known in the manufacturing environments. For instance, a jaw chuck (Figure 1a) has a correct general shape of its body (real jaw chucks which are used e.g. in lathes have cylindrical bodies) and general orientation of jaws is also correct in that case. In the case of tool holder (Figure 1b) the shapes and locations of elements are correct but they have no detailed geometrical features. Also, a screw-nut and a wrench models (Figure 1c) have some similarities to the real objects, however, the worst result was obtained in the case of a wrench geometry in authors' opinion which is not sufficiently similar to the real wrenches and also there is no thread generated on the screw, and the screw's head has only a general shape correctly generated. The gear wheels (Figure 1d) have a cylindrical shape which corresponds to the real gears and their position (one to the other) is correctly generated but there is no details such as teeth, central hole necessary to place a wheel on a shaft, etc.

The testing phase indicated that at this stage of LLM models development chats are able to create FreeCAD-related Python code that can generate simple objects but their correctness may depend on the prompting phase and still cannot propose very detailed and production-ready CAD models.

In the next stage of the research authors decided to create more structured prompts in order to ask about more detailed geometrical and functional properties of the generated solid models.

The Table 1 was designed to provide the detailed description of each assembly. The indicators presented in the Table 1 were developed to create the final more detailed prompt. Each prompt linked to one assembly contained descriptions from all the rows in the column entitled *Description.* They were copied to the ChatGPT^{TM} in order to obtain the Python language-based code. The codes, similarly to the previous attempts were run in the FreeCAD. At this stage authors focused on three different assemblies: a vise, a screw-nut assembly and a jaw chuck assembly.

Table 1. Indicators involved in the prompts

Designation	Indicator name	Description
D	Dimensions	Dimensions of parts and assembly such and length. height, thickness, etc.
L	Layout and constraints	Description of location of parts within the assembly, e.g. how one part is placed within the assembly.
N	Number of elements	Number of single elements, mainly if the number of similar parts is greater than 1.
G	Geometry of elements	Description of geometry such as its shape.
F	Functionalities	Description of functionalities such as how single parts influence the exploitation of the entire assembly, for instance: description of movements of parts or the applications linked to specific machined parts, etc.

The following paragraphs present the prompts developed on the basis of indicators in the table and used to generate 3D models. In some cases correcting prompts were also sent to the ChatGPT™ to achieve a better result compared to the result obtained on the basis of the first prompt. These correcting prompts are also presented in the subchapter.

Exemplary prompt for a vise

Generate a code which will be used in the Python console in the FreeCAD software and will help to create an assembly of the vise.
(D) *The vise shall have a length of 200 mm, a width of 80 mm and a total height of 80 mm.*
(L) *The vise shall have a base element to which 1 stationary and 1 movable jaw are fixed. The stationary jaw is secured by the use of screws on one side of the base, while the movable jaw is guided by the use of guides located on the base and on the jaw. Moreover, a screw is fixed to the movable jaw at one of screw sides and this screw goes through a nut which is fixed to the base on the opposite side (opposite to the stationary jaw location) of the base.*
(N) *5 solid models are included within the assembly in total.*
(G) *The base has a shape of a cuboid – a block. The jaws have the same shape as the base. The nut has a shape of a block with a threaded hole parallel to the screw axis. The screw has a shape of a cylinder.*
(F) *The jaws shall secure a machined part. The screw rotation around its axis shall move the movable jaw along the base of the vise, however, the nut is a stationary element.*

Correcting prompt: *The screw shall go through the center of the nut, be parallel to the longer side of the base. The nut shall have a threaded hole.*

The result obtained in the FreeCAD software is presented in the Figure 2.

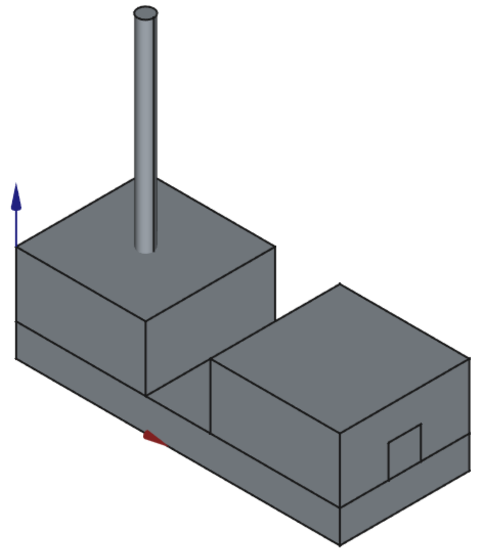

Fig. 2. An AI-supported vise model obtained in the FreeCAD software

Exemplary prompt for a screw-nut assembly

Generate a code which will be used in the Python console in the FreeCAD software and will help to create a screw-nut assembly.
(D) *The screw-nut assembly shall have a length of 100 mm in total (total length of the screw including its head). A diameter of the screw shall be 12 mm. The same diameter shall be created for threaded hole of the nut. The head of the screw shall have a height of 10 mm.*
(L) *The nut shall be screwed onto the screw.*
(N) *Number of solid models equals two – the first one is the screw model and the nut model is the second one.*
(G) *The head of the screw shall have the shape of a regular hexagonal prism. The nut shall also have the shape of a regular hexagonal prism.*
(F) *The final model represents the screw-nut assembly. The screw is screwed onto the nut.*

Correcting prompt 1: *Nut and screw shall be separate parts.*
Correcting prompt 2: *Nut shall be located in the center of the screw.*
Correcting prompt 3: *Nut shall have thicker wall.*

The Figure 3 presents the result obtained in the FreeCAD software.

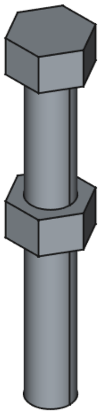

Fig. 3. An AI-supported screw-nut assembly obtained in the FreeCAD software

Prompt for a 3-jaw chuck

Generate a code which will be used in the Python console in the FreeCAD software and will help to create a 3-jaw chuck assembly.
(D) *A chuck diameter is 150 mm and its height is 80 mm. Each of three jaws has a length of 40 mm, a height of 20 mm and a width of 20 mm.*
(L) *The three jaws lay on the face of a cylinder representing the chuck body. They are placed every 120 degrees around the axis of the chuck.*
(N) *The number of solid models in the assembly is four – body and three jaws.*
(G) *The body of the chuck is a cylinder. The jaws are cuboids.*
(F) *The jaws may be translated to the axis of the chuck. This enables to clamp a machined part.*

The Figure 4 presents the result. In fact, after sending a few correcting prompts (6 in total) that were following the abovementioned main prompt, the result presented in the Figure 4 was the best one compared to the 3-jaw chucks existing in the real manufacturing environments.

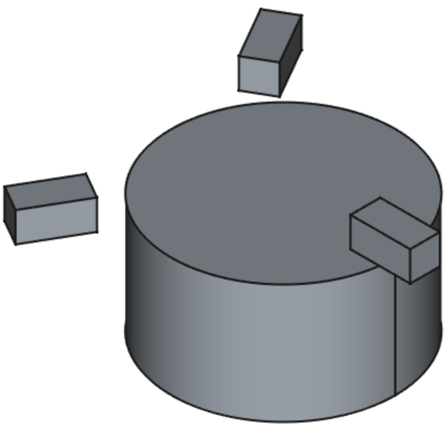

Fig. 4. An AI-supported 3-jaw chuck model obtained in the FreeCAD software

It can be stated now that a fully correct result of AI-aided modelling is still not reached. Only in the case of the screw-nut assembly the correct result was reached (please refer to the Figure 3).

The entire testing phase proved that some very general shapes and dimensions are generated according to designer's needs and ideas. In this context further improvements are necessary to be added. Authors also verified generated models' dimensions required in the prompts by using existing measure tool of the CAD environment and it can be stated that they were generated correctly as requested in the prompts.

In the conclusion of this subchapter, it can be stated that the required general shapes of separate elements in the assemblies are generated correctly, and the dimensions are also properly created, however, the position of elements or features (e.g. the threaded hole in the case of the nut of the vise) should be improved. Moreover, no detailed features (e.g. thread on the screw) were generated. In fact a person who writes a prompt does not know the result that will be created by the ChatGPT™ and this is quite disappointing for a designer but clear from the scientific point of view. In authors opinion this observation may define the nature of human and artificial intelligence collaboration concept. In this context, the final results (final outputs) of any intelligence work cannot be precisely predicted even if they are expected. However, AI-created (or human) outputs may be improved by other designers or other AI-based tools.

The currently verified concept presented in the previous sentence, proposed after completion of the studied examples during the testing phase, shifted the research to the next step that is focused on further improvement of initial results.

2.2. Improvement phase

This subchapter presents further methodology on the steps that can be completed in order to obtain the detailed models that meet requirements of designers.

Several possible approaches are possible to be applied and tested:

1. Further correcting prompts designing and expecting the better final results after a text-based talk with the ChatGPT™.
2. Manual (based on a designer actions) correction of the Python language-based code.
3. Manual correction of the best of all obtained 3D models using existing CAD tools in the FreeCAD or another CAD software.

Authors selected one type of the part (3 jaw chuck) and presented how a designer can proceed. Similar approaches can be applied in the case of other assemblies.

2.2.1. Further correcting prompts application

Further correcting prompts may lead to selected improvements of models geometry. Authors requested the ChatGPT™ for further improvements such as creation of T slots or more detailed geometry of the jaws. However, they found it very difficult to obtain the detailed final model efficiently. A designer shall consider if an effort put into the design by the use of prompts is effective or maybe another approach can be used. One of the results obtained after the correcting prompts application is presented in the Figure 5. It can be stated that positioning of geometrical features is reachable, however, it also requires additional workload which can be replaced by the direct modelling using CAD tools.

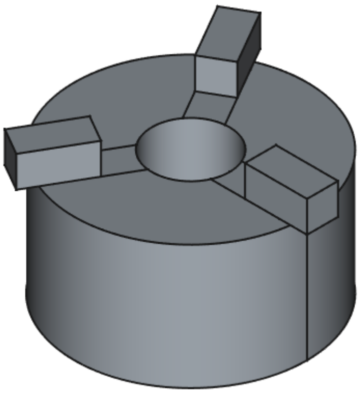

Fig. 5. An AI-supported 3 jaw chuck CAD model improved by the use of correcting prompts

2.2.2. Manual correction of Python language-based code

Manual correction of the Python code seems to be another possible solution. Nevertheless, looking from the perspective of a CAD tool accessibility and workload for a designer, in authors' opinion, the better results can be reached easily by the use of approach presented in point 2.2.3. However, in some cases selected values or single rows of the Python code may be changed to improve the result. There is also a good practice to ask in the prompt for additional comments to understand better and faster how every single command works. From the perspective of improvement phase, if a designer decides to rewrite the entire Python code because the basic one does not meet requirements, it means that there is a necessity to design the part from the beginning using the Python nomenclature.

2.2.3. Manual correction of 3D models in CAD environment

This approach enabled to complete detailed changes and the final obtained result is probably the best if a model is compared to chucks existing in the real manufacturing environments. In this case the AI helped to obtain simple cylinder and a block that were redesigned by the use of CAD tools. The final result is presented in the Figure 6. Authors have not implemented all the details, however, they presented that manual redesign is easier to be implemented in the case of detailed part's features. In this context, existing (traditional) CAD design methodology and tools still are very efficient.

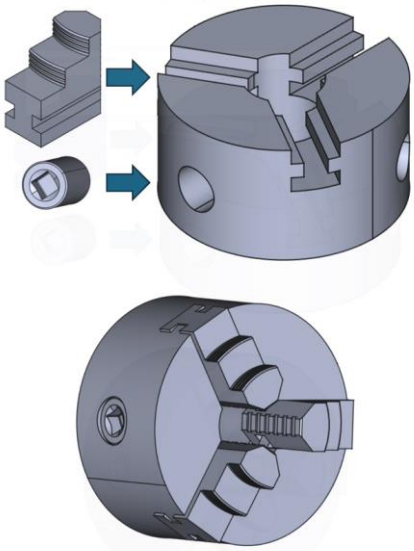

Fig. 6. A 3 jaw chuck CAD assembly model improved manually by a designer by the use of traditional CAD tools

3. Automated design of simple parts of the same type

The usage of part libraries in modern CAD environments is a standard procedure leading to faster creation of complex assemblies by using standard parts such as bearings, screws, rings, etc. Due to the best results (previously verified) regarding AI-aided creation of simple geometries such as a screw-nut assembly, authors developed that concept and studied if the ChatGPT™ can create a useful Python-based code that can be applied in the FreeCAD environment in order to generate a set of simple solid models in the chosen folder saved on the PC hard drive.

The basic prompt was as follows:

Generate code for the FreeCAD software Python console that will create hexagonal head screws having thread diameter of 12 mm. Length of the screws shall start from 10 mm to 500 mm long, in 10 mm increments. In total 50 screws shall be created. Let each solid model be saved in a folder entitled 'screws' created on the D drive of the PC.

Figure 7 presents the screw having the total length of 500 mm displayed in the FreeCAD environment after implementation of the Python code generated by the ChatGPT™ and selected from the created library. Verification of requested dimensions (requested by authors) by the use of existing CAD tools has also a positive output. Both length and diameter of screw were correctly generated. The Figure 8 presents the final code generated in the ChatGPT™ environment that is a result of the basic (first) and the second prompt. The first attempt caused an error and the ChatGPT™ explained that App.pi and related trigonometric functions (cosine and sine) were incorrectly referenced. The second attempt ended up correctly and 50 solid models were saved in the indicated folder on the drive of used PC.

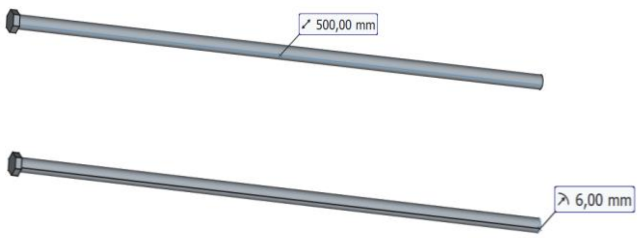

Fig. 7. Exemplary screw having length of 500 mm and diameter (2 x radius) of 12 mm – FreeCAD preview

The code enabled to create 50 solid models in ‘screw’ folder located on local drive. It is presented in Figure 9. Authors decided to generate by the ChatGPT™ support also 10 additional nuts having different heights ranging from 5 to 55 mm. The final result of prompting and generated Python code implementation in the FreeCAD environment is presented in the Figure 10 (An exemplary nut having diameter of threaded hole 12 mm and total length of 55 mm is presented).

These two examples indicated that using the ChatGPT™ in the case of libraries that consist of simple 3D models at this stage of LLM development can be successfully applied in the CAD libraries development and applications. It shall be also stated that the verified libraries of standard parts are also a very reliable solution when special parts are assembled in CAD environments and AI-support may play a significant role in that area.

```
import FreeCAD as App
import Part
import os

# Create the folder to store screws
output_folder = "D:/screws"
os.makedirs(output_folder, exist_ok=True)

def create_hexagonal_head_screw(thread_diameter, length):
    # Create a new document
    doc = App.newDocument()

    # Create the shaft of the screw
    shaft = Part.makeCylinder(thread_diameter / 2, length)

    # Create the hexagonal head
    hex_radius = thread_diameter * 0.75
    hex_thickness = thread_diameter * 0.5
    hexagon = Part.makePolygon([
        App.Vector(hex_radius * cos(angle), hex_radius * sin(angle), length)
        for angle in [i * 2 * App.pi / 6 for i in range(7)]
    ])
    hexagon_face = Part.Face(hexagon)
    hexagon_solid = hexagon_face.extrude(App.Vector(0, 0, hex_thickness))

    # Combine shaft and hexagonal head
    screw = shaft.fuse(hexagon_solid)

    # Add the screw to the document
    doc.addObject("Part::Feature", "Screw").Shape = screw
    doc.recompute()

    return doc

# Parameters
thread_diameter = 12  # mm
min_length = 10  # mm
max_length = 500  # mm
step_length = 10  # mm

# Generate screws
for length in range(min_length, max_length + 1, step_length):
    doc = create_hexagonal_head_screw(thread_diameter, length)
    file_path = os.path.join(output_folder, f"hex_screw_{length}mm.FCStd")
    doc.saveAs(file_path)
    print(f"Saved: {file_path}")

print("All screws generated and saved.")
```

Fig. 8. The final Python code used for screw library development created by the ChatGPT™ on the basis of prompt

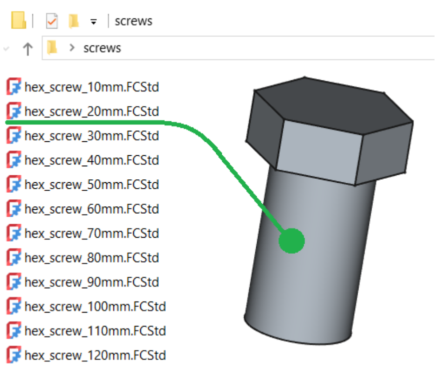

Fig. 9. Print screen of screw library in folder on the D drive of PC (Folder in Windows 10™ by Microsoft operating system)

Fig. 10. Exemplary nut – FreeCAD preview

4. Conclusions

The study focused on the analysis of CAD design of assemblies supported by the use of artificial intelligence. In authors opinion, the AI-based modelling of fixture assemblies is a promising approach having a great potential in the future of manufacturing. Automation of such design processes will probably help to decrease the overall design time. However, at this stage of AI developments there is still a need for improving LLM models in order to obtain better output code. Authors share a similar opinion often presented by researchers and top developers in the area of AI indicating that there is a need for prompting research. However, the analysis presented possibilities of the FreeCAD software combined with the ChatGPT™ by OpenAI and helped to obtain general information about the expected complexity of models that could be generated by the use of AI in a way that requires only minor changes to the basic CAD models. It also presented that a quite simple geometries can be correctly generated at this stage.

Moreover, creating 3D CAD models libraries is a very promising concept that can be simply and directly implemented in CAD designers work.

The work presented the results that extend an overall knowledge in the area of assembly design and possible applications of LLM models for CAD design at this stage of their development.

General attempts (based on authors observation) that have been made to make a quantitative comparison of research results by comparing the time of generating CAD models using AI and traditional methods are strongly linked to the authors skills. The effectiveness of using AI depends on the details of the prompts too. Authors suppose that the more accurate they are, the greater the efficiency of generating CAD models. The problem concerns their proper formulation and a lack of knowledge regarding their interpretation by the LLM. For this reason, in authors opinion, it is difficult to determine the time it will take to build a CAD model using AI. In the case of manual (traditional) modelling it mostly depends on CAD software and also on a CAD user skills. The study has shown that AI can support designer's work regarding creation of large number of repeatable solids, with simple and similar shapes. It definitely saves designer's time. Manual modelling in this case would certainly be time-consuming. While prompting methods are being improved and LLMs too, the designers should expect better LLMs efficiency and accuracy of AI-generated models. This will be the subject of further research together with application of other software tools (e.g. Python-based environments using dedicated Python libraries without the typical CAD software usage) and implementation in the real CAD project. Moreover, the results obtained in various chats may also be compared and discussed.

Acknowledgments

This work was developed within the project Teamwork-based Education and Digitalization as an Approach for the Interdisciplinary Engineering Training: https://erasmus-plus.ec.europa.eu/projects/search/details/2024-1-PL01-KA220-HED-000257156 – Funded by the European Union.

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Foundation for the Development of the Education System. Neither the European Union nor the Foundation for the Development of the Education System can be held responsible for them.

The results of this paper will be used in the international trainings for students developed within the abovementioned project. They also contribute the research-based education.

Artificial intelligence (AI) usage statement

The generative AI (ChatGPT™ by OpenAI) was used only to generate a Python code on the basis of prompts created by authors. Generated Python code lines were run in the FreeCAD software. All sentences in the paper were developed by authors without AI assistance, excluding the fact that we checked "a" and "the" articles in authors' prompts and translated a few words from Polish into English by the use of the ChatGPT™.

References

Badagabettu, A., Yarlagadda, S.S., Farimani, A.B. (2024). *Query2CAD: Generating CAD models using natural language queries* (arXiv:2406.00144). arXiv. https://doi.org/10.48550/arXiv.2406.00144.

ChatGPT by OpenAI, chatgpt.com (accessed: 29.12.2024).

Deng, H., Khan, S., Erkoyuncu, J.A. (2024). An Investigation on Utilizing Large Language Model for Industrial Computer-Aided Design Automation. *Procedia CIRP*, *128*, 221–226. https://doi.org/10.1016/j.procir.2024.07.049.

FreeCAD, www.freecad.com (accessed: 29.12.2024).

Hunde, B.R., Woldeyohannes, A.D. (2022). Future prospects of computer-aided design (CAD) – A review from the perspective of artificial intelligence (AI), extended reality, and 3D printing. *Results in Engineering*, *14*, 100478. https://doi.org/10.1016/j.rineng.2022.100478.

Kapsalis, T. (2024). *CADgpt: Harnessing Natural Language Processing for 3D Modelling to Enhance Computer-Aided Design Workflows* (arXiv:2401.05476). arXiv. https://doi.org/10.48550/arXiv.2401.05476.

Li, X., Sun, Y., Sha, Z. (2024, November 13). *LLM4CAD: Multi-Modal Large Language Models for 3D Computer-Aided Design Generation*. ASME 2024 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. https://doi.org/10.1115/DETC2024-143740.

Sarker, I.H. (2022). AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems. *SN Computer Science*, *3*(2), 158. https://doi.org/10.1007/s42979-022-01043-x.