



CAD BASED PRODUCT DESIGN: A CASE STUDY

Panagiotis Kyratsis¹, Evaggelos Gabis¹, Anastasios Tzotzis¹, Dimitrios Tzetzis², Konstantinos Kakoulis¹

¹Western Macedonia University of Applied Sciences, Department of Mechanical Engineering and Industrial Design
Kila Kozani, GR50100, Greece

²International Hellenic University, School of Science and Technology, 14km Thessaloniki - N. Moudania, 57001, Themi,
Greece

Corresponding author: Panagiotis Kyratsis, pkyratsis@teiwmg.gr

Abstract: It is well known that the design process of a product is one of the most important tasks that must be concluded prior to production. Even after a product is released to the market, it is certain that it will undergo some design changes during its evaluation stage. Thus, design engineers must be prepared to make the necessary changes as fast as possible, in order to successfully upgrade their product and recirculate it to the market. This perception leads the teams of engineers and designers to automate certain procedures that regard product design, manufacturing and relevant processes. The present paper deals with the development of an application, with the aid of the Application Programming Interface (API) of a commercially used CAD (Computer Aided Design) system. The selected software is SolidWorks™ which makes use of Visual Basic™ event driven programming language. The idea of this application is to provide the user with a simple and easy to use platform that can automate the design process of a high profile product such as a bicycle. It is a product with many different aspects i.e. size, frame style, color, wheels, handlebar, saddle, etc.

Key words: API, SolidWorks™, CAD based manufacturing, automated design

1. INTRODUCTION

Nearly 80% of the design activities have a routine nature. That is the reason for the increased demand on CAD based tools and applications that can automate standard routine activities. These activities regard either small and simple assemblies or larger ones with more complex structure. In all cases, the main aim is to save time and offer high quality products as fast as possible.

2. LITERATURE SURVEY

Throughout literature there are reports about many tools that were developed with the aid of different API systems. Some researchers developed applications that lean towards automatic design and manufacturing, while others built specialized tools for specific purposes like artificial measuring, machining simulation, robot simulation and more. Zbiciak et al.

prepared a software application with functionality that allows designing and modelling of spur and helicoidal involute gears in an accurate way by using involute curves with eleven involute points [1]. Kyratsis et al. worked on CAD based design and manufacturing of mechanical systems and developed the application named Cylin3Der that provide the user with a simple form that acts as a platform for the design process of a pneumatic acting mechanical system [2]. It is based on a set of parameters that takes into account the ISO 15552 norms. Similarly, Tzivelekis et al. developed an application that automates both the design and manufacturing processes of impeller type geometries by incorporating certain methods and equations for better approximation of the curved blades [3]. Tzotzis et al. created a collection of tools that can be easily accessed with the help of a simple user interface [4]. These tools provide various functions such as the automatic design of a mechanical system, the automatic creation of the mechanical drawings for each component of the designed system and the extraction of the assembly's properties that can be stored to a Microsoft Excel™ file for further study. The GENgine original software system by Haba and Oancea is dedicated to the design and digital manufacturing of the air-cooled single-cylinder engine block and the corresponding cylinder casing by integrating phases of CAD, CAPP (Computer Aided Process Planning), and CAM (Computer Aided Manufacturing) [5]. Another interesting use of the API is the development of a simulation system of a robotic hand by Tsagaris et al. The robotic hand has the shape of a real human hand and incorporates moving and rotating joints in order to successfully simulate the alphabet using the sign language [6]. The hand prototype is designed in Autodesk AutoCAD™, while the moving features were accomplished using Autodesk Inventor™. Kyratsis et al. developed the DRILL3D software routine by using the API of a modern CAD system. DRILL3D creates parametrically controlled tool geometries and by using

different cutting conditions, achieves the generation of solid models for all the relevant data involved during the drilling manufacturing process (drilling tool, cut workpiece, undeformed chip etc.) [7]. Another software tool that allows user to generate different types of holes, on parallelepipedic solids, starting from different references as edges, vertexes, known points on the solids, or different types of entities which are placed on the solids was developed by Badan and Oancea [8]. Pescaru et al. presented a software tool that deals with automatic design of a footwear product type and also describes the steps for designing the products with a high level of customization [9]. Additionally, Oancea and Haba worked on a new computer tool, which allows the user to obtain the technological manufacturing sequences, cutting data for each process from the manufacturing sequence and finally to assist at the simulation of each process. The tool is very useful in the integration process of CAPP with CAM system for rotational parts [10].

The present paper makes use of a CAD based API in order to show that one key advantage of using it, is the ability to build simple and easy to use tools, that can serve various purposes, when repeatable and time-consuming tasks are involved. It deals with the development of an application that can be used for automating the design process of a bicycle using the API of SolidWorks™.

3. CASE STUDY

The case study that follows is the automatic design process of a bicycle, which is a product that can undergo many design changes throughout its life cycle and thus involves a lot of exhausting optimization activities for the designers and engineers that need to be minimized or maximized.

3.1 Workflow

The workflow of the present application is based on a simple input – output philosophy. First of all, the user inputs all the necessary data, as seen in Figure 1, into an easy to use interface that is divided into five tabs.

The first tab contains a textbox for the data input, whereas all other tabs contain radio buttons that the user can click in order to make a selection. Next, the user can press the command button so that the design process of the product can begin. When the process ends, the output is a fully assembled and parametrically designed model of a bicycle that comprises a set of basic parts, which contain certain attributes i.e. material, texture and color. The aforementioned attributes are applied on the part models during their design process. In addition, a rendered model of the bicycle is produced by using the previously mentioned attributes for illustration purposes.

The produced models are saved in the user specified path that can be changed through a simple text box,

which is contained in the user form. Additionally, the user can specify the template path, also through a similar text box in the design form.

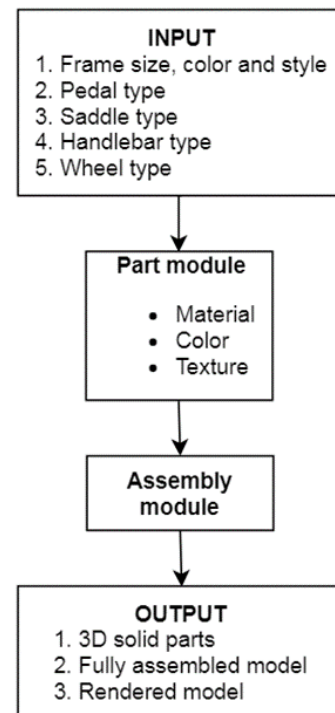


Fig. 1. Data flow of the application

The programming structure of the application is based on the principals of Visual Basic™. This means that upon triggering the application, it must first connect to SolidWorks™ and then make the declarations of the required variables. It is very important for the application to declare all the required variables and reserve memory space at the beginning, in order to avoid conflicts.

With the variables set, the steps of the design process are straightforward; at first the part module becomes enabled with an aim to permit the application to open a new part document and select a sketch plane. Subsequently, the sketch of the first part is created and then with the help of commands such as extrude, the complete 3D solid model of the part is produced. This process takes place several times until all parts of the bicycle are produced and simultaneously all part models are saved in the predetermined path.

Finally, the application enables the assembly module, opens a new assembly document and performs a series of mate commands, in order to assemble the complete bicycle. The flow of the design process for the spur gear component is illustrated in Figure 2. This flow is generally applied, which means that all parts are created with the depicted series of actions.

With all solid models of the bicycle's components completed, the next step is to create the final assembly of the bicycle. This is done via the assembly module of SolidWorks™ API. The assembly process of a wheel is shown in Figure 3.

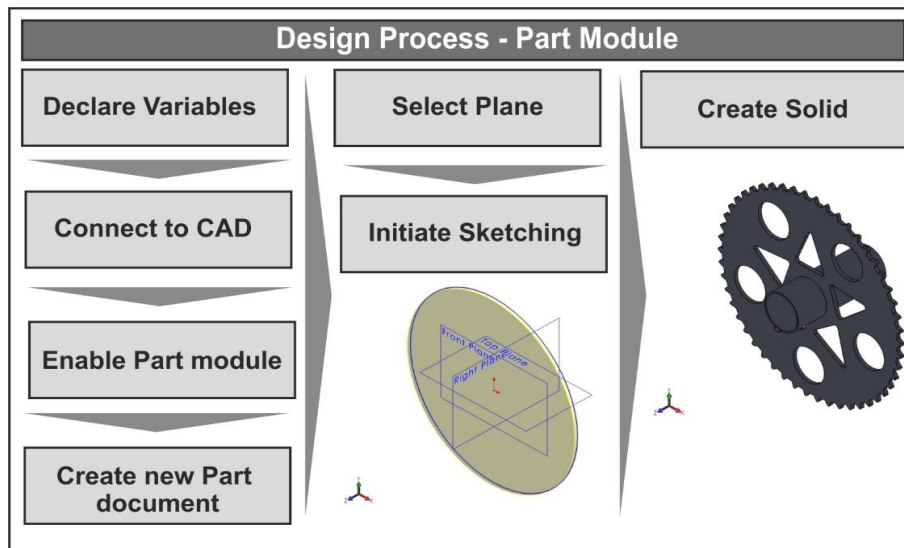


Fig. 2. Part design process of spur gear

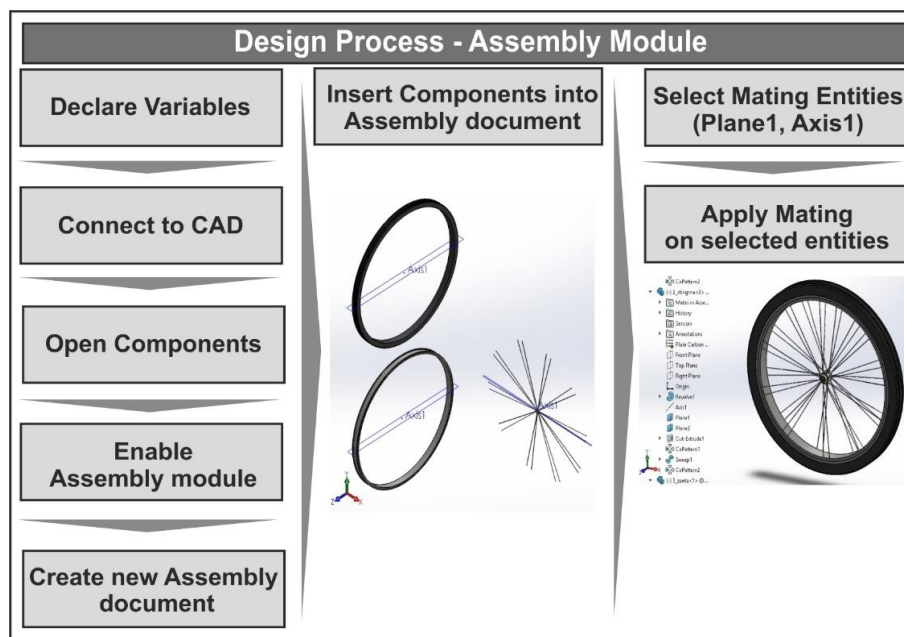


Fig. 3. Assembly process flow of a wheel

The assembly process flow is similar to the part design flow, because the first steps are exactly the same. The application must declare the variables, connect to the CAD system, open all the components that are relevant to the assembly and then enable the assembly module, which resides at the higher ranks of the API hierarchy, in order to create a new assembly document. The next steps are related to the assembly procedure and are totally different from the last steps of the part design process. As the components are loaded to the computer memory, the application is now able to insert them into the newly created assembly document. Finally, the application selects the appropriate mating entities like planes, axis, etc. and then performs a series of mate commands with the selected mating entities. During the example of the wheel assembly, the selected entities were the plane named Plane1 and axis named

Axis1 of each of the three involved components.

The reason that both the part design and the assembly design processes share common steps is that they belong to the same interface of the API object model, which is MODELDOC2 as seen in Figure 4. Moreover, from the depicted diagram it is clear that SLDWORKS is the top interface in the API, which provides direct and indirect access to all other interfaces exposed in the SolidWorks™ API. That is why the application has to connect to SolidWorks™ before any other action is taken. All interfaces in the API can be properly accessed if the hierarchy is respected. For example, MODELDOC2 is an interface that can be accessed after connection with SLDWORKS API object is established and allows access to SolidWorks™ documents such as parts, assemblies, and drawings. Similarly, PARTDOC and ASSEMBLYDOC can be accessed after the

connection with MODELDOC2 is established and together are the API objects that enable the use of the part document functions and assembly document functions respectively.

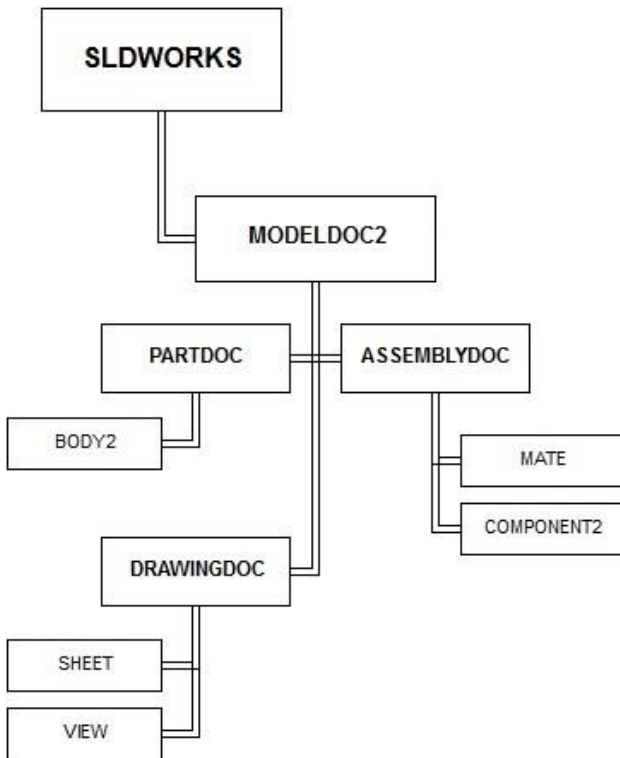


Fig. 4. Part of SolidWorks™ API object model

3.2 Interface

Figure 5 illustrates the interface arrangement of the application. The first tab deals with the most basic parameter of the bicycle, which is the frame size. User can choose a frame size between twelve and twenty inches that is considered as market standard size and it depends on the rider's height, gender, age and other characteristics.

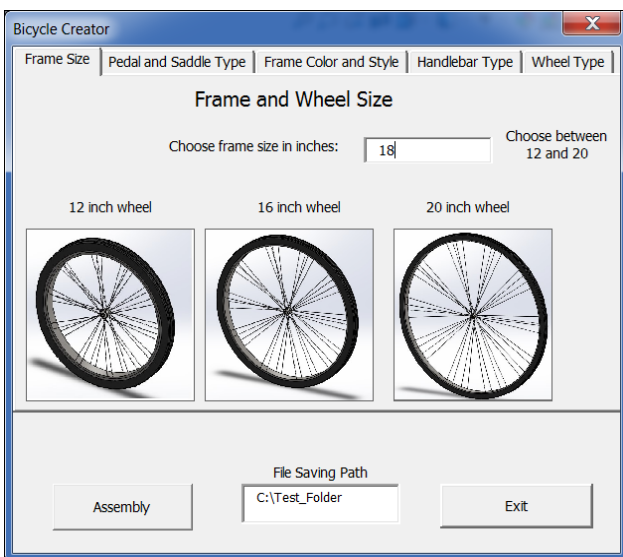


Fig. 5. First tab of the user interface

On the second tab, user can select the preferred pedal and saddle type. Two choices are available for both pedal and saddle type that can be activated via the corresponding radio button. Additionally, the tab contains images that describe the pedal and saddle types as seen in Figure 6.

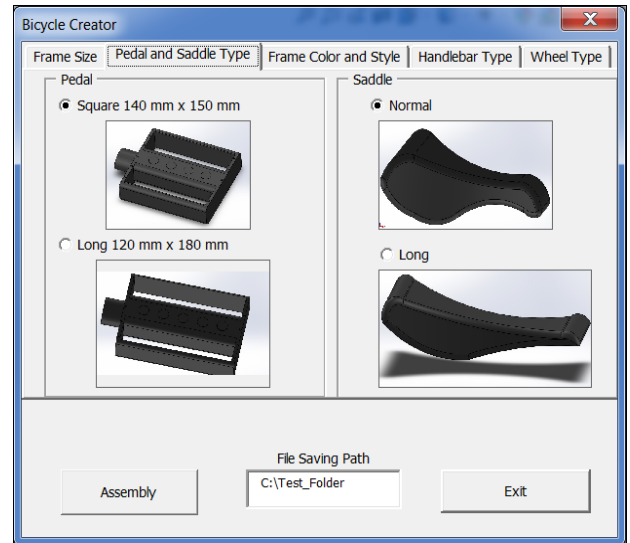


Fig. 6. The pedal and saddle selection tab

Third tab contains the options for frame color and style. Users can choose between male and female style by clicking the corresponding radio button. The styles that are available are common and meet the design standards of most manufacturers. Of course, there are many more styles commercially available but the philosophy is similar, thus it would be unnecessary to include more styles at this moment. Finally, users can choose a color of preference from a pool of five basic colors by selecting the name of the color from the appropriate selection box, as seen in Figure 7.

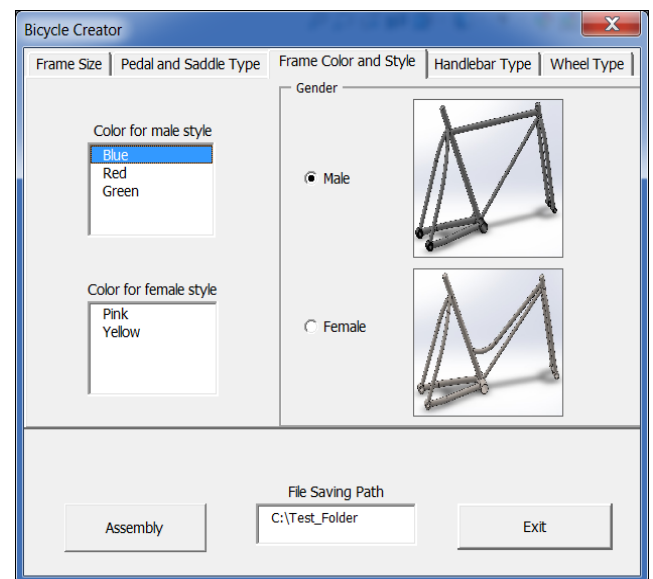


Fig. 7. The frame color and style selection tab

The next tab contains two options for the handlebar; a normal horizontal type and a more sports oriented type with curved edges. For now two types of handlebars are sufficient as the discrepancies between handlebars of different manufacturers are limited. Like all tabs, images accompany the radio buttons for better visualization of the bicycle components.

The fifth and final tab provides the user with two options for the wheels. Users can choose between wheels with regular spokes or wheels with rims that can be found on more advanced bicycles. Furthermore, user can select whether to add helping wheels to the bicycle or not, in case that the bicycle is intended to be used by beginners. In order to conclude the design process, the user has to press the “Assembly” button just after making all the required design selections. The application makes use of the input data in order to build the parts and then assemble them into a parametric assembly model. An example assembly model of a twelve inch frame bicycle is illustrated in Figure 8.



Fig. 8. The fully assembled model of a small frame bicycle

3.3 Design parameters

For the application to build the bicycle model it is first needed to calculate the key dimensions of the bicycle based on the chosen frame size. These dimensions are embedded into the code as variables that are linked to each other through mathematic formulae. Most of the dimensions are considered as design parameters but not all. Some dimensions were fixed in order to limit the code lines and because there are so many manufacturers with so many different bicycle models and types that is almost impossible to record every bicycle design. Due to the fact that the application is intended to be kept as simple as possible, the only parameter that the user has to input is the frame size. All the other design parameters are calculated based on the frame size. The most basic design parameters are summarized in Table 1.

Table 1. Basic design parameters

Design Parameter
Wheelbase
Fork length
Fork angle
Wheel outer diameter
Frame tube diameter
Seat tube angle
Bracket length
Disk radius
Handlebar length
Bottom bracket to fork top

3.4 Calculations

In addition, the application can easily calculate the wheelbase and all other important dimensions like trail, reach and stack by using international design standards that relate the frame size with the dimensions of the rest of the bicycle. Due to the vast number of bicycle size and style options, application includes some of the most common combinations of sizes and styles that can be found on manufacturers around the globe. An example regarding key metrics for a twelve inch frame, that is considered a small frame, is given in Table 2. User can expect the size of the bicycle with the help of the images of three sample wheels that are present on the first tab of the application, thus making the decision easier.

Table 2. Key metrics for a small sized bicycle frame

Dimension Name	Metric
Trail	71mm
Head tube Angle	71.3°
Seat tube Angle	68.9°
Bottom bracket Height	289mm
Wheelbase	1030mm
Effective top tube	615mm
Reach	370mm
Stack	639mm
Seat - Bottom bracket	281mm
Fork Length	420mm

Figure 9 depicts the most important dimensions of a bicycle that affect bicycle’s performance and rider’s comfort and provides a brief explanation of each of the dimensions. These characteristics are used to create a metric table like Table 1, which is necessary for the proper construction of the bicycle.

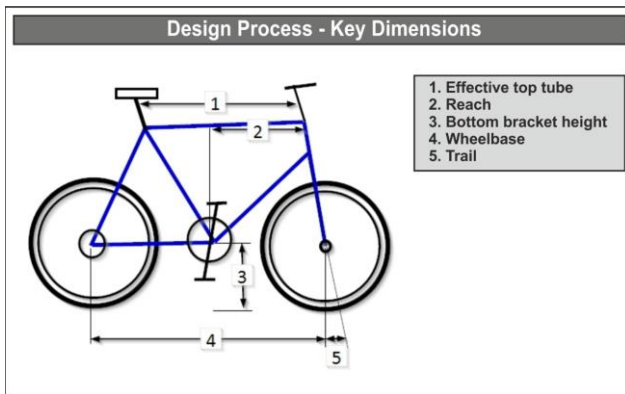


Fig. 9. Bicycle key dimensions

4. CONCLUSIONS

Overall, the use of a CAD API leads to increased productivity and better allocation of work time. This can be achieved by developing a tool or an application with the help of the API that each time suits the needs of a specific activity, without having to spend too much time on coding or learning to program with more powerful but complex programming languages. Especially when the developed tool or application is able to perform a number of desired tasks unattended and with minimal inputted parameters, then it is considered an invaluable asset to designers and engineers everyday work.

5. FEATURE WORK

In future works, it is anticipated to enrich application's current design combinations with additional sizes, styles and colors so that the application can produce more design schemes and to add a calculator that will dimension the bicycle by considering the ergonomics and not directly the bicycle's frame size. It is also expected to expand application's usability by adding automatic creation of drawings and technical documents, as well as automatic code generation for machined components.

6. REFERENCES

1. Zbiciak, M., Grabowik, C., & Janik, W. (2015). *An automation of design and modelling tasks in NX Siemens environment with original software-generator module*. In IOP Conference Series: Materials Science and Engineering, **95**(1), 012117, IOP Publishing.
2. Kyratsis, P., Tzotzis, A., Tzetzis, D., & Sapidis, N. (2018). *Pneumatic cylinder design using CAD-based programming*. Academic Journal of Manufacturing Engineering, **16**(2), 107-113.
3. Tzivelekis, C. A., Yiotis, L. S., Fountas, N. A., & Krimpenis, A. A. (2017). *Parametrically automated 3D design and manufacturing for spiral-type free-form*

- models in an interactive CAD/CAM environment*. International Journal on Interactive Design and Manufacturing (IJIDeM), **11**(2), 223-232.
4. Tzotzis, A., Garcia-Hernandez, C., Huertas-Talon, J. L., Tzetzis, D., & Kyratsis, P., (2017). *Engineering applications using CAD based application programming interface*. In MATEC Web of Conferences, **94**, 01011, EDP Sciences.
5. Haba, S. A., Oancea, G., (2015). *Digital manufacturing of air-cooled single-cylinder engine block*. The International Journal of Advanced Manufacturing Technology, **80**(5-8), 747-759.
6. Tsagaris, A., Stampoulis, K., & Kyratsis, P. (2018). *Hand finger gesture modeling in advanced CAD system*. Computer-Aided Design and Applications, **15**(3), 281-287.
7. Kyratsis, P., Bilalis, N., & Antoniadis, A. (2011). *CAD-based simulations and design of experiments for determining thrust force in drilling operations*. Computer-Aided Design, **43**(12), 1879-1890.
8. Badan, I., & Oancea, G. (2010). *Software Tool Used for Holes Modeling in AutoCAD Environment*. Proceedings of the Latest Trends on Engineering Mechanics, Structures, Engineering Geology, pp. 152-156.
9. Pescaru, R., Kyratsis, P., & Oancea, G. (2017). *Software tool used for automated design of customizable product*. In MATEC Web of Conferences, **137**, 06003, EDP Sciences.
10. Oancea, G., & Haba, S. A. (2016). *Software Tool Used in CAPP/CAM Systems for Rotational Parts*. Scientific Bulletin Series C: Fascicle Mechanics, Tribology, Machine Manufacturing Technology, **30**, p.75.

Received: April 05, 2019 / Accepted: December 20, 2019 / Paper available online: December 25, 2019 © International Journal of Modern Manufacturing Technologies