

AN INTEGRATED RBR FIXTURE DESIGN SYSTEM

Oleg Mihaylov,¹ Galina Nikolcheva,²

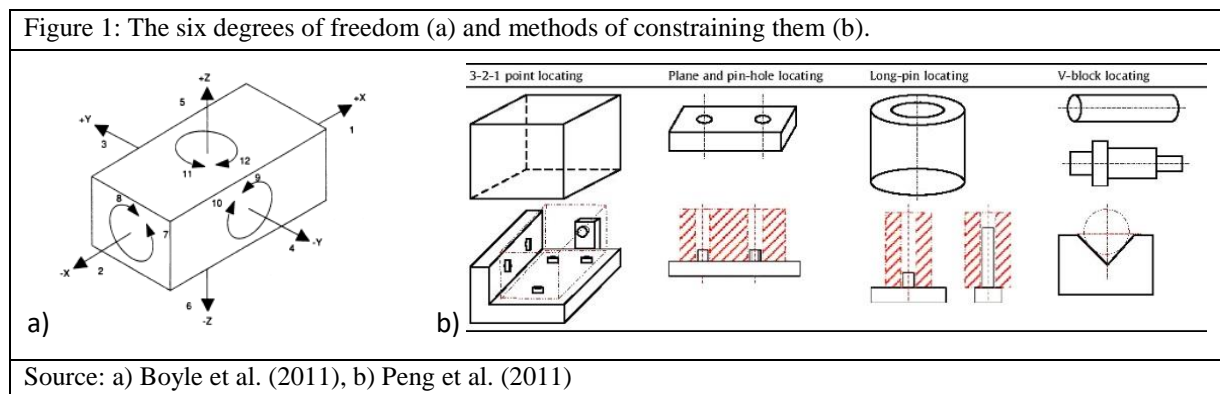
Abstract: Traditionally fixture design processes are costly and time-consuming when carried out manually. To try to reduce the negative features, we are creating a computer-aided fixture design system, integrated in the SolidWorks' environment. This system uses rule-based reasoning in the form of If-Then-Else rules, incorporated in its VB.NET code. This paper presents our progress in the form of an interactive add-in program for locating parts. This program generates modular fixtures for locating prismatic or cylindrical parts, using rules and mathematical equations. Its usage can shorten the time needed for determining the type and position of every module of the fixture, their assembling, and inspection.

UDC Classification: 004.42; **DOI:** <http://dx.doi.org/10.12955/cbup.v5.1091>

Keywords: Fixtures, computer-aided fixture design, solidworks, add-in

Introduction

A fixture is a work-holding or support device used in the manufacturing industry – job, batch, and mass production. Fixtures are used to position the workpiece in a specific location and orientation, and support it during machining, ensuring that all parts produced using the same fixture will maintain conformity and interchangeability. These tasks require restricting the six degrees of freedom of the workpiece (Fig. 1a) and there are a variety of locating methods that can facilitate this such as: the 3-2-1 locating principle (6 points location), plane and pin-hole locating, long-pin locating, V-block locating (Fig. 1b).



The costs associated with fixture design and manufacture can account for 10%–20% of the total cost of a manufacturing system (Bi et al., 2001). These costs relate not only to fixture manufacture, assembly, and operation but also to their design. Approximately 40% of rejected parts are due to dimensioning errors that are attributed to poor fixture design (Nixon, 1971). Hence there are significant benefits to be gained by reducing both the design time and the costs associated with fixturing.

Modular fixturing systems are the most widely used flexible fixturing systems in manufacturing. They are based on the standardization of their components and they are designed as groups of pre-manufactured standard elements and units with relatively tight tolerances. Those elements can be assembled in variety of different fixtures for locating and clamping a variety of parts with different geometries, sizes and fixturing requirements. Once all the needed processes are completed the fixtures can be disassembled and reassembled in other completely different configurations (Zhu & Zang, 1990). This is the modular systems' biggest difference and biggest advantage over the dedicated fixtures as the latter are usually scrapped after completing their task. Generally, there are two types of modular fixture systems – T-slot-based and dowel-pin-based, each with their advantages and disadvantages.

¹ Faculty of Industrial Technology, Technical University of Sofia, Bulgaria, omihaylov@tu-sofia.bg

² Faculty of Industrial Technology, Technical University of Sofia, Bulgaria, ginic@tu-sofia.bg

Fixture design system – interface and operation

SolidWorks is a CAD system allowing the use of an application programming interface (API), which allows the automation of many designing and analytical processes for both single parts and assemblies. This is achieved by creating a program with one of the supported programming languages (C++, C#, Visual Basic etc.).

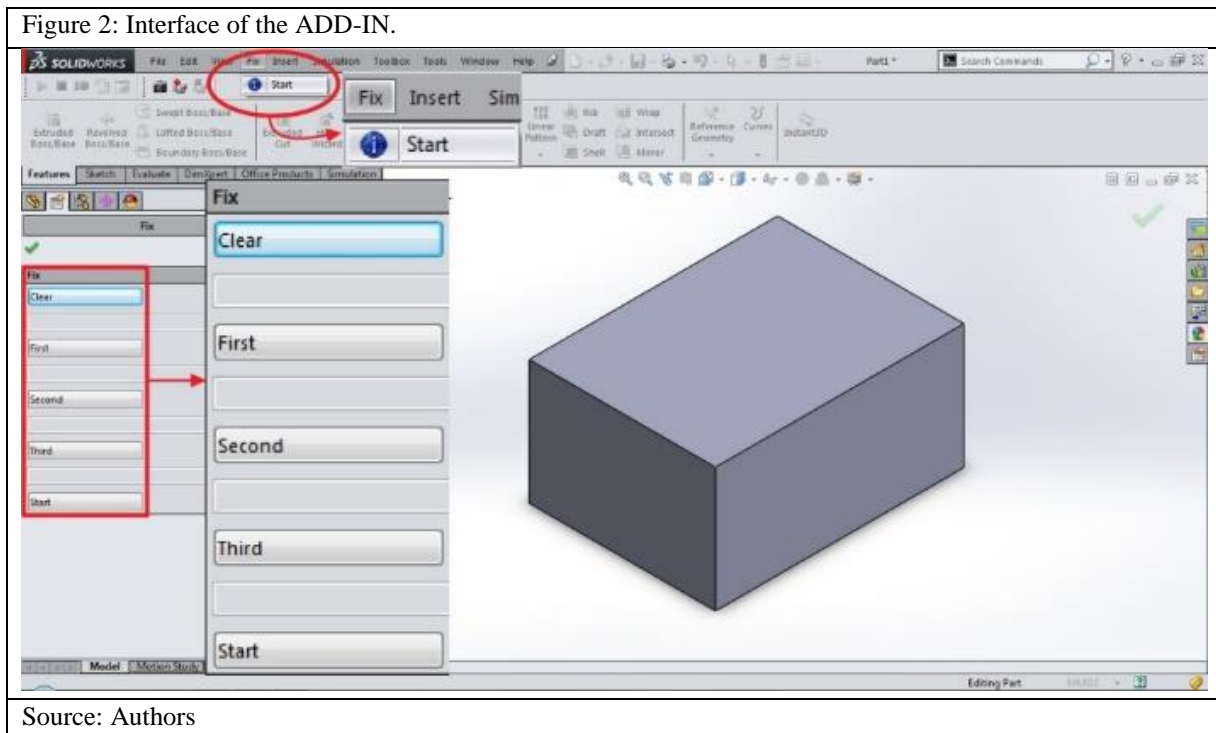
An add-in integrated in the SolidWorks' environment was developed by using the SolidWorks' API and the VB.NET programming language. Its task is to automate the fixture design process for locating both prismatic and cylindrical workpieces. The locating is by one of three types – the 3-2-1 locating principle, the plane and pin-hole locating, or the V-block locating. These three options is what gives this add-in greater versatility than other programs which are concentrated only on one method or one type of parts. The automation of the design process is partial because the program needs the user to select locating surfaces, in other words the program is interactive.

The working principle of the add-in is as follows:

With an opened file of type "PART" the user starts it from the Toolbar which opens a new Property Manager Page on the left side of the window (Fig.2). This page consists of:

- button "Clear" – deletes all names and additional features (axes, planes, points etc.);
- 3 buttons for surface selection – the user selects a surface then presses a button to confirm the selection and to check if the surface isn't already selected;
- 4 text fields – for displaying information to the user (e.g. The surface is selected, The surface is not acceptable, etc...);
- button "START" – it starts the automated creation of an assembly file.

Figure 2: Interface of the ADD-IN.



Source: Authors

The user selects a surface and then presses one of the three buttons for selection to let the system collect data from it. The system's first task is to determine what type of surface is selected – planar, cylindrical or other – and to decide which locating method to use. After that the system collects data on the shape and size of the surface and calculates the positions of the locating elements. The user has to wait for the system to finish its task for each surface before he can select the next one, depending on the surface's shape and complexity this may take from a few seconds up to 1-2 minutes. After the selections, the user lets the system create the fixture assembly with the button "START."

Determining the positions of the locating elements

With the location of the parts it is crucial for the locating elements to be correctly positioned. Those positions directly depend on the surface's shape (rectangle, circle, polygon), its dimensions, and if it's solid or not, i.e.: are there holes, steps, etc... The program needs to "understand" what portion of the part it can use for locating with the help of the user and the set of rules. This allows the program to find other surfaces in the same plane without the user selecting all of them. The rules are using the SolidWorks' options, available through the application programming interface (API). After the user selects a surface and presses a button, the program determines its orientation by extracting its normal vector data and gets the coordinates of the end points. Then the program checks every surface of the part for others with the same orientation and in the same plane. If there are such surfaces their end points' coordinates are added to those of the selected surface.

After the comparison is concluded, a new sketch is created on the selected surface. The coordinates gathered from all end points are transformed to the local coordinate system of the sketch and the maximal and minimal values for each axis are determined - $x_{1\ max}, x_{1\ min}, y_{1\ max}, y_{1\ min}$. Those values are used for the calculation of a center point O_1 :

$$(1) \quad x_{10} = \frac{x_{max} + x_{min}}{2}, \quad y_{10} = \frac{y_{max} + y_{min}}{2}.$$

The locating points are going to be calculated in relation to this center point in such manner that they will imitate the holes pattern on the base plate. Next the distance between the minimal and maximal value for each axis is calculated:

$$(2) \quad \Delta_x = x_{1\ max} - x_{1\ min}, \quad \Delta_y = y_{1\ max} - y_{1\ min}$$

$$(3) \quad k_x = \left\lfloor \frac{\Delta_x}{0,050} \right\rfloor, \quad k_y = \left\lfloor \frac{\Delta_y}{0,050} \right\rfloor.$$

The coefficient k shows the multiplicity of the distance Δ to the step of the holes on the base plate (50mm) rounded down to an integer. Those coefficients are used for the calculation of the coordinates for the location elements so the round down guarantees that they will be between the endpoints of the surface. The coordinates are calculated as follows:

$$u_{11} = u_{12} = u_{10} - k_u * 0,025; \quad u_{13} = u_{10} + k_u * 0,025;$$

$$(4) \quad v_{11} = v_{10} - k_v * 0,025; \quad v_{12} = v_{10} + k_v * 0,025;$$

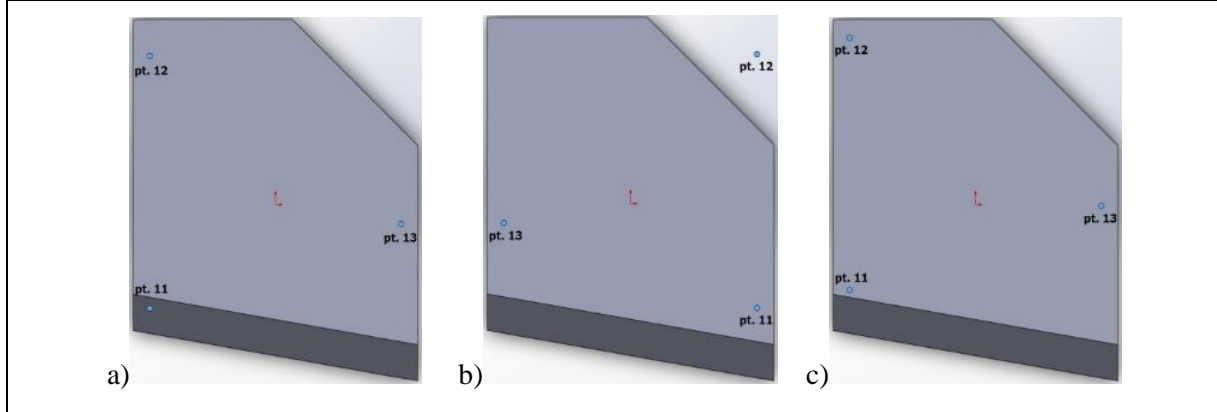
$$v_{13} = v_{11} + \left\lfloor \frac{0,5 * (v_{12} - v_{11})}{0,050} \right\rfloor * 0,050.$$

The values for u_{10}, k_u, v_{10} and k_v are taken from equations (2) and (3) depending on which value of Δ is greater – for $\Delta_x \geq \Delta_y \rightarrow u = x, v = y$, and for $\Delta_x < \Delta_y \rightarrow u = y, v = x$. The same logic is applied for the values of $x_{11}, x_{12}, x_{13}, y_{11}, y_{12}$ and y_{13} . With the calculated coordinates the program creates points in the sketch and closes it. Next there are checks on whether the points lie on the surface (or surfaces). The task of those checks is to avoid placing a locating element in a hole, a groove or outside the surface. If at least one point is not on the surface (or surfaces) (Fig.3.a) the program implements one or series of corrections aiming to get all points on the surface.

The corrections can be the mirror reversal of the points to one axis (Fig.3b) or a single or multiple displacement of all the points along one or both axes at a distance equal to the portion of the free space f (Fig.3c), i.e. from (2) and (3):

$$(5) \quad f = \frac{\Delta - (k * 0,050)}{2}.$$

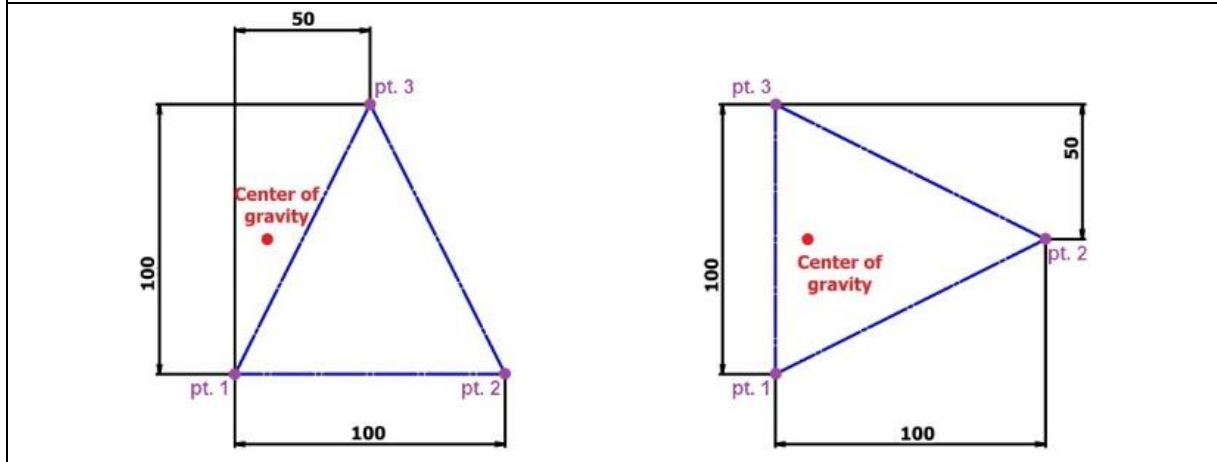
Figure 3: Locating points – a) calculated coordinates; b), c) with corrections.



Source: Authors

Other corrections are recalculations of the coordinates with coefficient $k-1$, with the possibility of repetition until $k = 1$ or swapping the coefficient u and v . The different corrections can be combined in various ways so after one type of correction, multiple corrections of different type can be executed.

Figure 4: Positioning the locating points according to the center of gravity.



Source: Authors

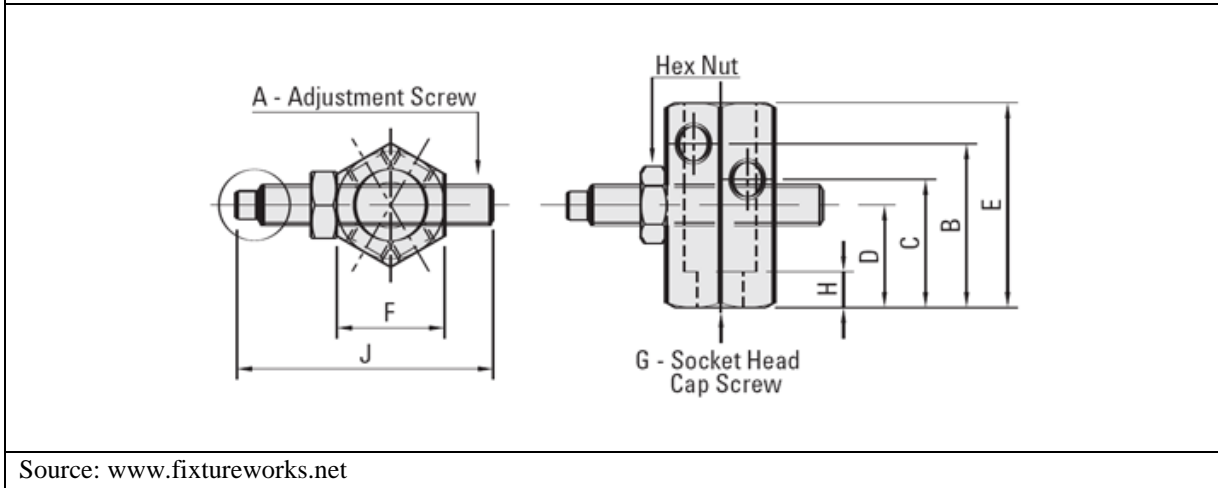
When all points are lying on the surface, the program checks if the center of gravity (its projection on the plane of the surface) lies in the triangle of the three points. If it does, then the determination of coordinates is completed successfully, otherwise additional corrections are implemented. The corrections consist of displacement of one or two points along one or both axes at distances which are multiples of 50mm (Fig.4.).

After the successful selection of the first surface (i.e. all checks are passed successfully) the user can make a selection for the second and third surfaces. For determining the locating points on the second and third surfaces the center point $O_1(x_{10}, y_{10})$ is projected onto their planes (thus ensuring the same symmetry in all planes) but the coordinates are determined differently than for the first surface. For the second surface the coordinates for one of the axes (the one parallel to the projection of the first surface) are calculated according to O_1 :

$$(6) \quad u_{21} = u_{20} - \left(\left\lfloor \frac{u_{2max} - u_{2min}}{0.050} \right\rfloor * 0.050 \right) / 2; \quad u_{22} = u_{20} + \left(\left\lfloor \frac{u_{2max} - u_{2min}}{0.050} \right\rfloor * 0.050 \right) / 2.$$

In (6) u_{21}, u_{22} are the coordinates on the first axis; u_{20} is the coordinate of the projected point O_1 on this axis. The coordinates on the second axis v_{21}, v_{22} are selected according to the selected adjustable stops (Fig.5.).

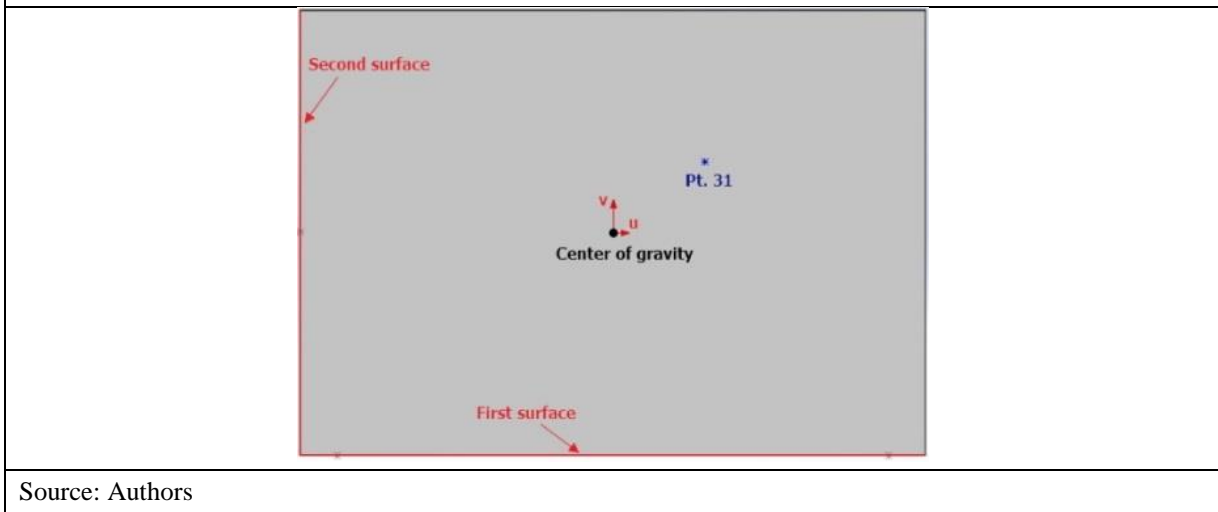
Figure 5: Positioning the locating points according to the center of gravity.



Source: www.fixtureworks.net

The height of the adjustable screw is selected so that the center of gravity (its projection on the plane) is between the locating points and the first surface's projection. The program creates a point with the calculated coordinates and starts the checks and corrections, similar to those for the first surface. The corrections can be mirror reversal or displacement of one or both points by 50mm on each of the axes. If necessary the locating points can be moved below the center of gravity.

Figure 6: Position of pt.31 in relation to the first and second surface and the center of gravity.



Source: Authors

For the third surface the coordinate u_{31} is calculated similar to u_{21} and u_{22} :

$$(7) \quad u_{31} = u_{30} \mp \left(\left| \frac{u_{3max} - u_{3min}}{0.050} \right| * 0.050 \right) / 2.$$

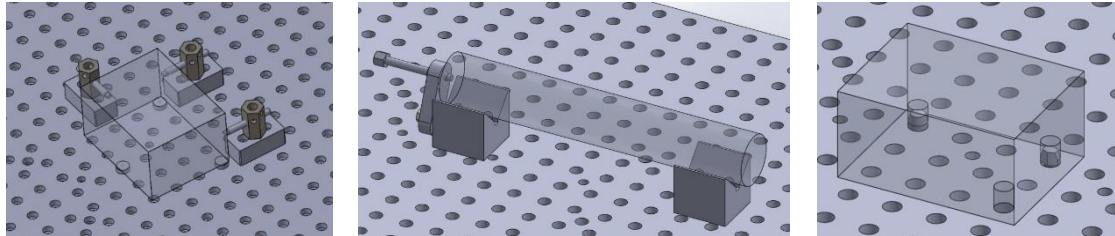
The symbol \mp in (7) is replaced with + or - so that the point of center of gravity is between the locating point and the second surface's projection (Fig.6.). For the second axis the coordinate of the point is selected the same way as v_{21} and v_{22} . The checks and corrections are same as for the second surface.

Assembly

Once the selection process is completed successfully the user can select the button "Start" and let the program create a new file of type "ASSEMBLY" (.sldasm). In this file, the program places the base plate as a fixed part and starts inserting and mating processes for all locating and supporting elements and the part. The program inserts the default configuration of every element and changes it according to the values of the v_{21} , v_{22} and v_{31} coordinates (for 3-2-1 locating) or the diameter (for the stopping elements with the V-bloc). If it's needed the adjustable stopper's body can be placed on raising elements for additional height. The need, type and dimensions of the raising elements are automatically selected by the system.

All elements are mated together at the calculated positions and the system concludes its work by leaving the new assembly for the user to interact with it – to inspect, to manually correct it if needed, to save the file with a selected name and in a selected folder etc. The system can design a locating fixture for one of three methods – 3-2-1 location, plane and pin-hole locating, or V-block locating (Fig.7.).

Figure 7: Assembled fixture with part.



Source: Authors

Conclusions

This paper reports the development of a fixture design system that is able to generate modular fixtures for locating prismatic or cylindrical parts. The system is integrated in SolidWorks and uses the SolidWorks' options and virtual environment to create an assembly of modular elements from a database. The system is interactive – it needs input information from the user in the form of a selection of surfaces. This information is passed through a series of If-Then-Else rules (Rule-Based Reasoning), compiled by the authors, for the accurate determination of the positions of the locating elements. The system automatically determines the needed modules (type, configurations, number) and their position on the base plate, and mates them into an assembly file.

A methodology for determining the locating elements' positions is developed, with rules and mathematical equations. The equations and the rules ensure the precise determination of the locating coordinates for the modules, and their further assembly in the new file. The methodology was developed for the 3-2-1 locating principle, but part of it is used in the Pin-hole locating method.

By using this program, the user can shorten the time needed to determine the type and position of every module of the fixture and in assembling them. The user can also use it for the visual inspection of the design without the need of assembling the fixture, or for comparing two types of locating (e.g. 3-2-1 principle and Plane and pin-hole locating) for comparing locating by different sets of surfaces. Work is currently being carried on completing the system with clamping and verification.

References

- Bi Z. M., Zhang W. J. (2001) Flexible fixture design and automation: review, issues and future direction. *International Journal of Production Research*, 39(13), 2867–2894.
- Boyle, I., Rong, Y., Brown, D. C. (2011) A review and analysis of current computer-aided fixture design approaches. *Robotics and Computer-Integrated Manufacturing*, 27, 1-12.
- Nixon F. (1971) *Managing to achieve quality*. McGraw Hill: Maidenhead.
- Peng G., Chen G., Wu C., Xin H., Jiang Y. (2011) Applying RBR and CBR to develop a VR based integrated system for machining fixture design. *Expert Systems with Applications*, 38, 26–38.
- www.fixtureworks.net
- Zhu Y., Zang S. (1990) *Modular Fixturing Systems: Theory and Application*. Machinery Press. Beijing.