XXI IMEKO World Congress "Measurement in Research and Industry"
August 30 – September 4, 2015, Prague, Czech Republic

SOFTWARE TOOLS AND TRICKS USED TO DEVELOP APPLICATION THAT EXPLAINS GEOMETRICAL TOLERANCING CONCEPTS

Marcin BERTA [1], *Zbigniew HUMIENNY* [2]

[1] Precision Machine Tools Factory "AVIA" S.A., Poland, marcinberta@gmail.com
[2] Warsaw University of Technology, Institute of Machine Design Fundamentals, Poland, zhu@simr.pw.edu.pl

Abstract – The paper demonstrates a new concept of teaching and training geometrical tolerancing fundamental and advanced principles with usage of computer simulations, as well as animations at a very large extend. The general concept, along with some programming solutions employed to develop an application *Geometrical Tolerancing* are presented. The application contains definitions, illustrations, plain animations, cartoon sequences and exercises for self-tuition that explain the ISO GPS tolerance system.

Keywords: geometrical tolerancing, tolerance indication, e-learning, ISO 1101

1. INTRODUCTION

The geometrical product specification and verification system – the ISO GPS system – developed in the ISO by Technical Committee ISO/TC 213 [1, 2] enables the unique definition of allowable deviations of an actual workpiece from its nominal geometry that do not deteriorate expected functional requirements. A tremendous amount of information on geometrical tolerancing is available in the ISO standards [3], however they are written in specific language and are not easy to understand for students or industry people involved in design, manufacturing and inspection processes. Most of the textbooks are well structured, however they contain only still drawings and more informative full-color pictures are very rare used. This research corresponds to the expectations of personnel competency in the geometrical tolerancing specifications and verification [4] as well as it takes into account recently reported results of effective application of computer aided learning [5].

2. STRUCTURE OF THE APPLICATION GEOMETRICAL TOLERANCING

The application *Geometrical Tolerancing* starts from the *Main window*, in which a user can click on one of the 14 *Tolerance symbol buttons*, *Datums button, Size button* or any *Modifier button* to open the *Case selection window* with the list of cases of its applications. The *Main window* of the application with the mouse cursor being dragged and held over the *Position button* is shown in Fig. 1. The click event opens the *Case selection window* (Fig. 2) where applications of the particular position tolerances, some with modifiers, with respect to single datum or datum system are listed.

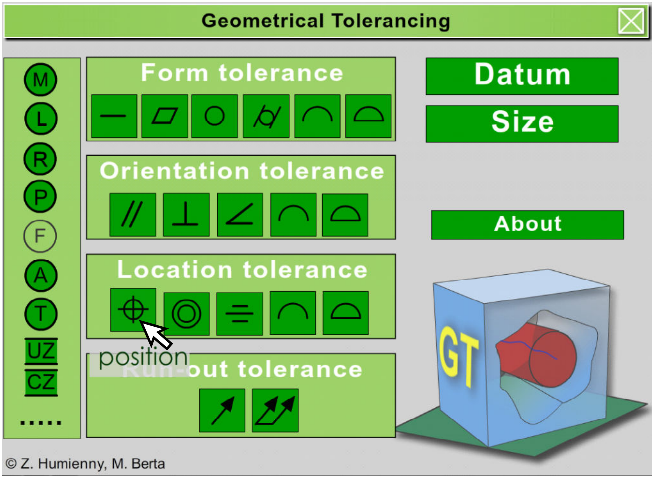

Fig. 1. *Main window* of the application *Geometrical Tolerancing* – the description of the *Position button* is displayed adjacent to the button because the arrow cursor is dragged and held over the button.

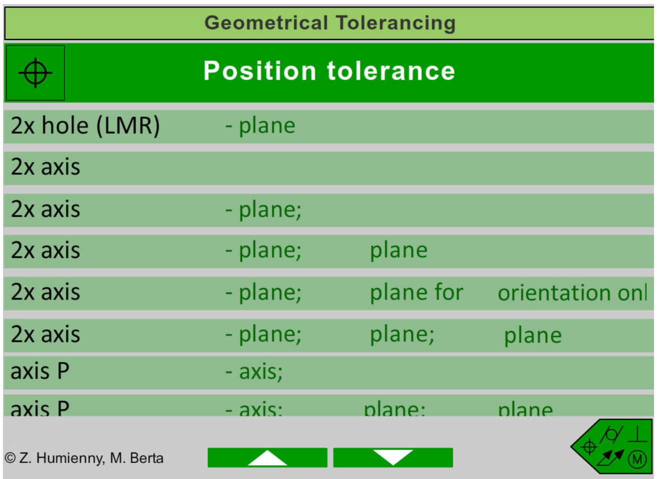

Fig. 2. *Case selection window* with *Line buttons* list of the position tolerance – current list contains 14 cases (available by scrolling).

Selection of the particular case by the *Line button* opens the *Definition window* (Fig. 3) with the relevant tolerance and the datum indicators attached to the geometrical features of selected workpiece. Any user after a moment can intuitively operate the application due to the authors' decision that only three window templates are employed in the application. The *Definition windows* for particular tolerances are used to present the key rules for the tolerance indication, as well as to explain how the tolerance zone is established and how the datum (datum system) constraints the spatial placement of the particular tolerance zone (Fig. 3). On the right edge of each *Definition window* the *Pull down menu* with four options (*Specification, 2D-drawing, 3D-drawing, Explanation*) is accessible. In each *Definition window* a user should trigger the *Explanation window* by the *Explanation button* (Fig. 3) to understand profoundly the meaning of a particular tolerance specification.

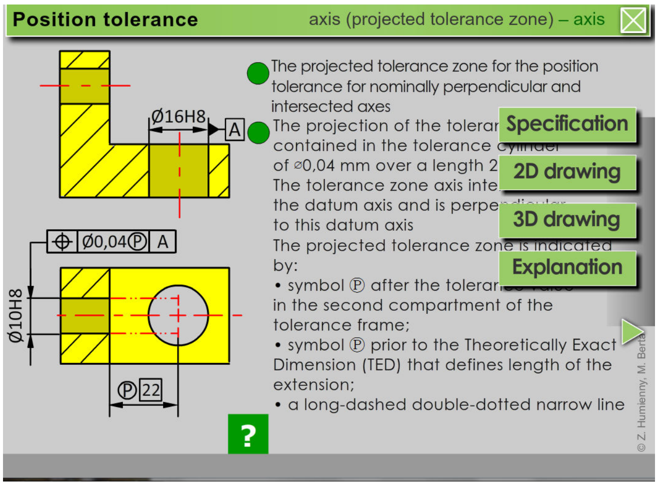

Fig. 3. *Definition window* for the position tolerance with modifier Ⓟ and single datum. The expanded pull down menu shown on the right edge guides a user to the next screens.

The on-screen notes and the context *Pop-up windows* are implemented in many definition windows. They are activated while the mouse cursor is dragged and held over the *Question mark icon* (Fig. 4) and used to explain in detail requirements defined by the specification, making it more easy to understand. The application of the *Pop-up window* is shown in Fig. 4 for the position tolerance of the projection of the cylinder inscribed into horizontal hole with respect to the datum system established by three datum features.

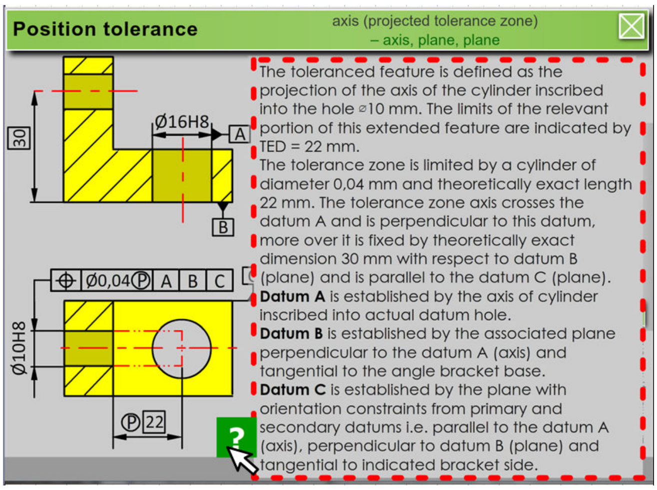

Fig. 4. *Definition window* for the position tolerance with modifier Ⓟ and three plane datum system with displayed *Pop-up* window used to explain how the datums in the system are established.

Each *Explanation window* (Fig. 5, 6) starts from an actual feature. Next, the *Explanation windows* are used, to show the tolerance zone(s) for the respective specification and/or to present inspection methods for selected tolerances. Each tolerance zone for toleranced integral or derived feature is clearly visualized – running translation and/or rotation of the tolerance zone shows its unlocked degrees of freedom – and finally the tolerance zone is shrunk or expanded to demonstrate the evaluation of a particular deviation. When applicable, the first click event on the *Forward button* that is available in the bottom right corner of every one *Explanation window* animates the explanation how the tolerance feature is established – in the case of the modifier Ⓟ the projection of the axis established by the inscribed cylinder is generated. Next click events on the *Forward button* animate the vivid explanation, how the tolerance zone is situated by the datum or datum system listed in the tolerance indicator.

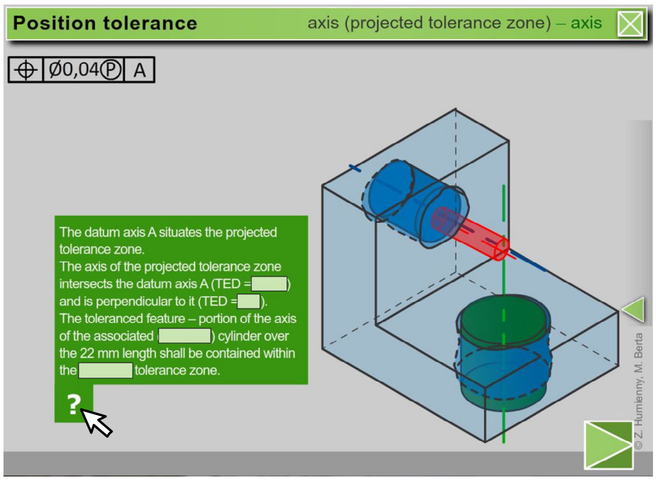

Fig. 5. *Explanation window* for specification from Fig. 3 – the fifth scene. The cylindrical projected tolerance zone for the projection of the horizontal hole axis is shown. The *Pop-up* window with short recapitulation and brief test is implemented in this last but one explanation window scene.

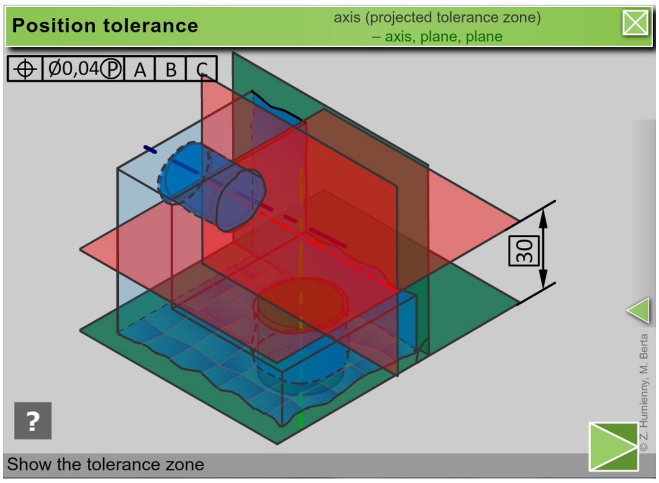

Fig. 6. *Explanation window* for specification from Fig. 4 – the seventh scene. The theoretically exact distance 30 mm indicates the theoretically exact location of the projected tolerance zone with respect to datum B.

According to ISO 10578 standard the projected position tolerance zone applies to external projection of the feature indicated in the drawing by symbol Ⓟ placed in the second compartment of the tolerance indicator after value of the position tolerance. In the application *Geometrical Tolerancing* the modifier Ⓟ is employed for tolerancing of crossed axes. Two cases of the position tolerance with this modifier (Fig. 3, 4) are discussed in the application to underline the role of datum(s) listed in the tolerance indicator. The tolerance is used to limit distance between actual axes that are nominally crossed axes. In both cases the projection of the axis of the cylinder associated to the toleranced hole (horizontal hole) shall be contained in cylindrical tolerance zone of diameter 0,04 mm over the length 22 mm. When one datum is specified (axis of the cylinder inscribed into vertical hole – Fig. 3) the axis of the tolerance zone meets the datum axis and the two axes form an angle 90°. The tolerance zone can freely float up/down and rotate around the datum axis that is suggestively visualized in the application by the zone movement in the *Explanation window* (Fig. 5). At the second specification (Fig. 4) these unconstrained degrees of freedom of the tolerance zone are locked by the position tolerance with respect to three plane datum system. The role of the datum order is visualized in the application by the sequence of dynamic scenes. Initially the cylinder is inscribed in the datum hole (animated "blow up" of the cylinder in the hole is shown) – its axis defines primary datum A. Next the plane derived from the secondary datum B – an associated plane that respects the orientation constraint (perpendicular to) from the primary single datum (axis) is shift up for the theoretically exact distance 30 mm. Finally the tertiary datum C – an associated plane which respects the orientation constraints firstly from the primary datum (parallel to) and secondly from the secondary datum is shift to include the axis of the vertical hole (primary datum). In this way the localization of the axis of the cylindrical tolerancing zone is fixed. The given description is a bit complicated even for people familiar with the tolerancing issues. So proposed visualization makes it much more understandable – that is undisputable advantage of the developed application. The Fig. 6 contains the seventh scene of the series of nine animated scenes that serve to visualize assessment of the position deviation (axes offset, "crossing deviation") in this *Explanation window*.

3. SOFTWARE TOOLS AND TRICKS

Visualization become more and more useful aid to creating educational materials that helps to understand difficult engineering problems [5]. There are many types of software that can be used to make valuable animations [6].

The authors decided to use Adobe Flash Professional. The Adobe Flash is often used by e-learning portals that help students in learning and achieving new skills accessible anytime and anywhere. An animation in the Adobe Flash Professional is based on *Timeline*. The Timeline in the Adobe Flash Professional is used to organize and control a document's content over time in layers and frames. Like movie film, the Flash documents divide lengths of time into frames. The layers are like multiple filmstrips stacked on top of one another, each containing a different image that appears on the stage. The major components of the Timeline are layers, frames, and the playhead [8]. Potentially four types of animations available in the Flash (motion tween, bone animation, shape tween and frame-by-frame animation) may be implemented in the application *Geometrical Tolerancing* to show efficiently and effectively the following aspects of the geometrical tolerancing:

- rules for specifications;
- explanations of the given specifications;
- verification methods.

Two Timeline techniques *motion tween* and *frame-by-frame* animation are implemented in the developed application *Geometrical Tolerancing* that is continuously improved and expanded for a couple of years [7]. Below some details are discussed. The tools used during the application development are presented on the case of the angularity tolerance of the plane with respect to the datum plane. The *Definition window* (Fig. 7) is available from *Main window* (Fig. 1) by the *Angularity Button* and then by selection of the respective *Line button* in the *Case selection window*.

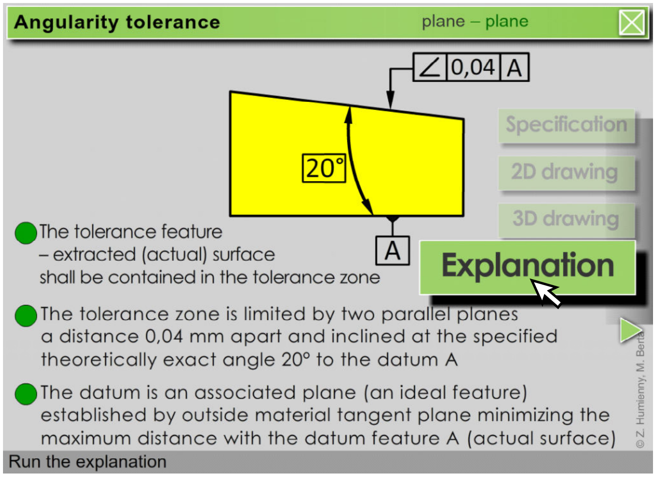

Fig. 7. *Definition window* for the angularity tolerance of a plane with respect to datum plane.

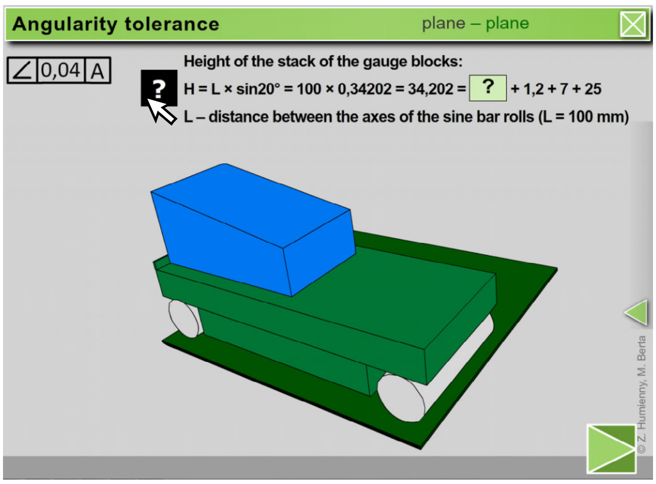

Fig. 8. *Explanation window* for specification from Fig. 7 – the first scene. The toleranced wedge is fixed on the sine bar plane. The rolls of the sine bar are on the surface plate. The height of the gauge block stack is calculated and the particular gauge blocks are determined.

The popular method of angularity deviation assessment with usage of classical measuring equipment (sine bar, gauge blocks, dial gauge, gauge stand and surface plate) is visualized in the application. The measured workpiece is fixed on the sine bar and then the bar is tilted to the surface plate by theoretically exact angle due to the gauge block stack placed between the bar roll and the plate. Thanks to that the measured surface is set parallel to the surface plate and angularity deviation may be easily find by the full indicator movement of the dial gauge fixed in the stand that is shifted on the plate.

The first scene of the *Explanation window* for angularity tolerance (Fig. 8) is released when a mouse cursor is held over *Explanation button* (Fig. 7) and then the mouse button is pressed and released (click event). This scene initially contains the sine bar placed on the surface plate. The toleranced wedge is set on the sine bar. The question is: *What is the height of the gauge block stack, that must be placed under sine bar roll?* The correct answer to this question is shown in Fig. 7, because the mouse cursor is dragged and held over the *Question mark icon.* The gauge blocks that form the stack are listed. It is worth to notice that the height of one gauge block from the stack is hidden. The application user should propose the answer and then she/he can check it by click event to disclose right answer covered by the rectangle. The right answer is 1,002 mm. The next scene of the *Explanation window* – scene in which the gauge blocks are wrung together to the stack – is released by *Forward button* available at the bottom right corner of the window. The subsequent scenes of the *Explanation window* will not be discussed from the application user point of view. These scenes will be used to show the tools used to develop the application *Geometrical Tolerancing*.

The *Frame-by-frame animation* is the simplest way to create animation which shows changes on a *Stage*. The Stage – the specific term for the Flash – represents the entire area where the Flash content is shown. For the application user the Stage means rectangular space in the Flash Player which is displayed in a web browser window where the application *Geometrical Tolerancing* appears. In each frame values such as position, color, shape, size of objects are changing and give the impression of a movement. This method of creating animation allows to have complete control of each frame. On the other hand this method increases the application file size. The Flash doesn't use the vector transformation, but stores information about all frames separately. Such type of animation is used in the application only for creating the 3D animations, because it is the simplest way to integrate files with animated graphics from two different programs. The Swift 3D which is the best tool to create a 3D solids and their movements may be integrated with the Flash which is handy software for a developer to create application with programmed features and a 2D animation. The Swift 3D has functionality that allows to export animation in .swf format. The Flash imports this file from the Swift as a frame by frame animation.

The Swift 3D is the standard software for the 3D simple modeling, rendering and animating. The environment of the Swift 3D is shown in Fig. 9. The main window of the program contains two Viewports ①. The drawn objects can be shown in the viewports from side that is suitable for an animator. It is very useful way to control location of the objects on the 3D scene. The *Hierarchy Toolbar* ② serves as a central repository for every object that exist within executed scene. Each object has properties that are peculiar to all objects. An animator can decide about the rendering and smoothing options, size and position in the 3D, point of rotation, mesh properties. What is very convenient during the application development is that the Swift allows to precisely determine which outlines should be shown or hidden. The material appearance can be chosen by an animator. It is possible to control how the object surfaces interact with light. The designed surface can be glossy or matt. That is important for an animation perception. The listed properties have an impact on the file size and time to completion of the render job. The authors decided to make all animations in a similar manner as plain paper drawings – without solid body modeling – so the size of files and preparing time are smaller. The Swift 3D has the timeline, similarly to the Flash, where an animator can make a simple animation of an object movement on the scene. There are several properties of an object that can be animated, and some of them are listed on the left of the time line ③.

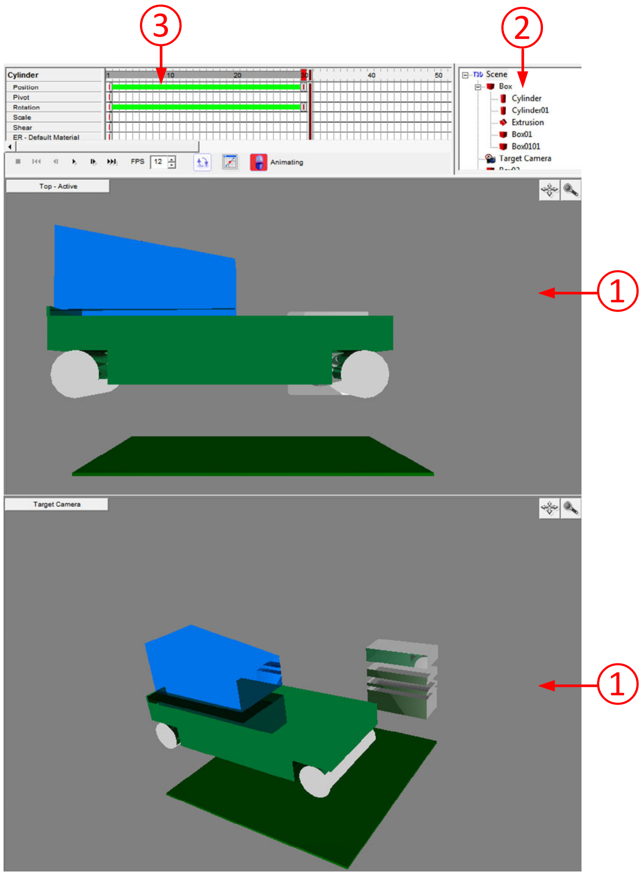

Fig. 9. The main window of the Swift 3D environment with two configurable viewports for viewing a scene.

The Swift 3D window of the rendered preview is shown in Fig. 10. It is used to verify whether entire animation is correct. The final result ② obtained from rendering process is displayed and may be observed frame by frame ①. A user can choose between two algorithms for an image rendering *Raster* ④ or lighter *Vector* ③. The authors decided to use the Vector. The Vector rendering reduce the size of output file and what is important the animation looks great. The file is saved frame-by-frame with usage of vector notation. The big advantage of the Swift 3D is that all animations are built all frames at a time. Much like a hand cartoonist created the old-time single flip book with image after image. Acceptable animation can be exported to flash file .swf by using the button *Export All Frames* ⑤.

The small web format (.swf) has become the standard for vector graphics on the web pages – the application *Geometrical Tolerancing* is designed for web browsers. This format is the low-bandwidth solution for creating animations and interactivity that may be displayed across multiple platforms and devices. The Swift 3D file .swf is a translation from a 3D to a 2D graphics for the Flash that is not supporting the 3D natively and is not able to tween the 3D objects between keyframes. The Swift 3D animation is stored with information what the Flash should display in each and every frame. In the other words, keyframes for every frame of an animation have to be exported from the Swift 3D.

When the final file prepared in the Swift 3D environment is exported to flash it appears in the Flash time line as the

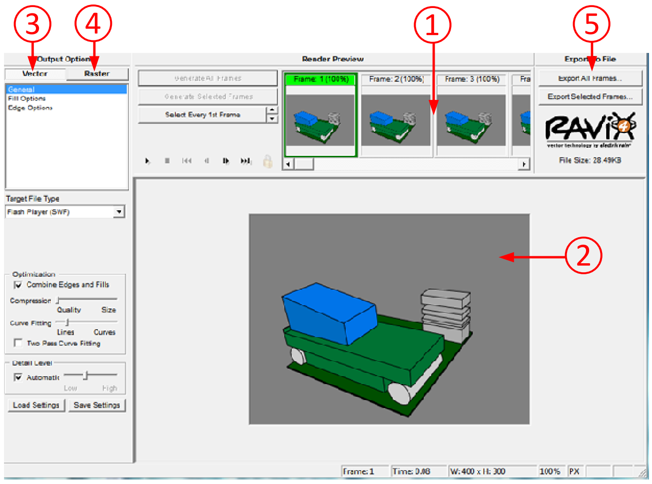

Fig. 10. The captured window of Swift 3D application when previewing rendering in order to visualize the wringing of the gauge blocks to get desired stack.

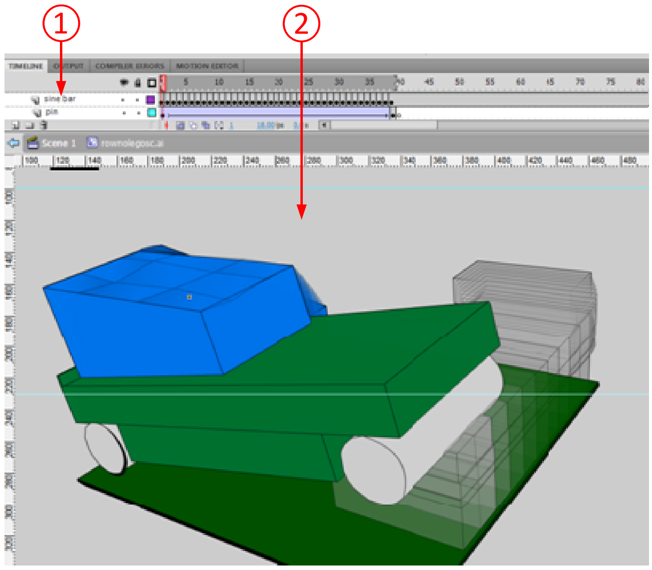

Fig. 11. The creation of the frame-by-frame animation. The frames of the sine bar are imported from Swift 3D.

animation frame-by-frame The creation of the second scene of the *Explanation window* included in the animation *Angularity tolerance plane–plane* is shown in Fig. 11. The animation is imported from the Swift 3D file and all sine bar frames ① are shown in one preview during creation of the stage ②. This animation is introduced to visualize tilting of the sine bar to the theoretically exact angle achieved by aligning the gauge block stack under the sine bar roll.

Using the other tool – the motion tween the authors generated the animated sequence to show how the dial gauge should be operated (Fig. 12). The motion tween is an animation type that allows to animate object using only the values for an object properties in start and end frame. In between frames values are calculated automatically by the Flash. This method involves the process of generating intermediate frames between objects to give the impression that first image smoothly transforms in to the next. In the discussed example the motion tween is combined with *Guide*. This tool allows to animate the dial gauge and related objects on the base of the stroke of the guide path. There is no need to put in any turning points between frames to change the direction of a movement. The creation of the animation using the motion tween and the path is shown in Fig. 12. The first Keyframe ① defines the initial position of the dial gauge and objects that are connected with the gauge. The layers ⑥ accessible in the animation toolbar contain objects which are moving with dial gauge. They are on the different layers because of the necessity to arrange transparency. The Flash doesn't arrange priority of a 3D scenes in a stack. An animator, with the layers usage, must decide which object overlaps the others i.e. which object is on the foreground and which are hidden. The second KeyFrame ② contains end location of the animated dial gauge and its stand. The tween frames ③ are automatically

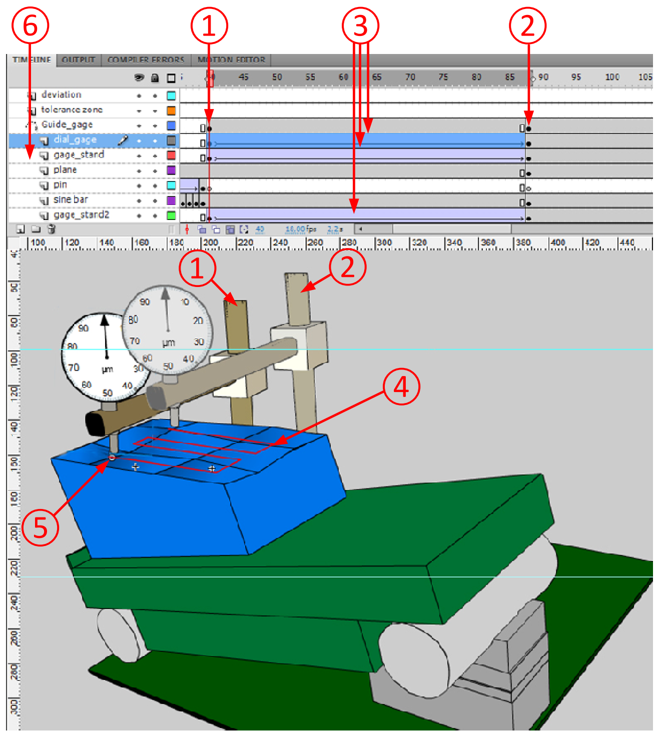

Fig. 12. Captured screen of the flash animation stage during creation of the angularity deviation measurement animation with the Flash tools motion tween and motion path.

generated by the Flash. The connection between animated object and path ④ is carried out by the control point ⑤ that refers to location of the object and the path.

The motion path is a especially designed line defining movement of twined objects. The Flash allows to draw an open line and define it as a trajectory for objects which are connected by the control point. The path line represents the target locations of selected objects during the timeline moves. The Flash keeps the control over a point location on the path and enables to fix any moment for an object to come out during an animation – not only in start and end points. The Fig. 12 was captured with usage of the multiple frames to show how it looks on the beginning of the animation and on the end of the animation.

The products of the authoring tools described above combined with usage of the *Forward Button* allow a user to go to the last but one scene in the *Explanation window* for the angularity tolerance of the plane with respect to the datum plane (Fig. 13). The readings of the dial gauge that represent the variation of distances from the points on the actual toleranced surface to the plate plane are displayed. The angularity deviation of the inspected wedge is equal to the readings difference. The displayed tolerance zone is established by two parallel planes a distance 0,04 mm apart and inclined at the theoretically exact angle 20° with respect to the datum A, that is simulated by the sine bar plane. The demonstrated tolerance zone is continuously animated in the loop to show that it is free to float – only its theoretically exact angle 20° should be kept.

The *Pop-up window* displayed due to the mouse cursor held over the *Question mark icon* (Fig. 13) is used to draw once more attention of the application user that specification of the angularity tolerance requires explicit indication of the theoretically exact angle between a tolerance feature and a datum. The click on the covering rectangle displays the hidden answer: *theoretically exact angle*. The covering rectangle is useful to verify whether a user of the application is familiar with this basic term.

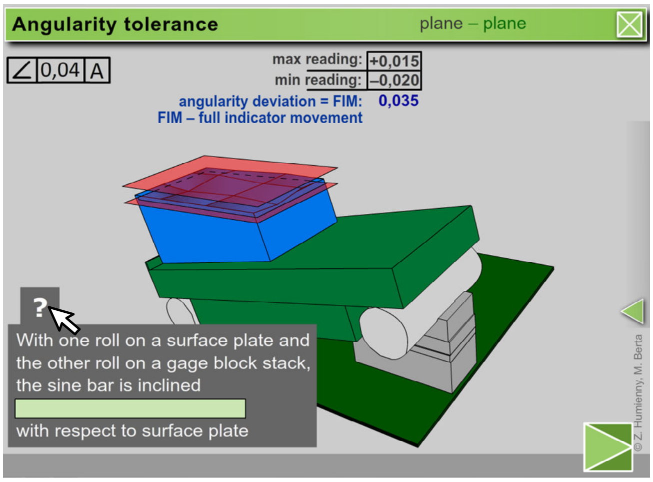

Fig. 13. *Explanation window* for specification from Fig. 7 – the forth scene. The tolerance zone is shown and the value of the angularity deviation is displayed.

The last fifth scene of the *Explanation window* is triggered by the *Forward button* available in the bottom left (Fig. 13). It is used to show how the angularity deviation is determined graphically. The planes that initially represent the tolerance zone are brought closer until they rest on the highest pick and the deepest valley of the actual toleranced surface. The distance between the planes fixed in this away determines the angularity deviation.

4. CONCLUSIONS

The animations are powerful and appropriate tool in the explanation of geometrical tolerancing concepts. Effect of the actual feature geometry variations is often difficult or impossible to observe directly in real parts due to the range of actual changes. Thanks to the animations that create virtual reality the feature variation may be significantly enlarged to emphasize the issue of the inherently imprecise manufacturing.

It was found during a number of vocational trainings performed mainly in the automobile and aviation industry that computer based training with usage of multimedia animation technology attracts the participants attention and significantly improves the learning efficiency.

The standardization is dynamic and never ending activity. The geometrical tolerancing standards are based on the improvements in manufacturing processes as well as on the development of measuring equipment and progress in measuring techniques. Currently the draft of the new edition of the standard ISO 1101 *Geometrical product specifications (GPS) – Geometrical tolerancing – Tolerances of form, orientation, location and run-out* is under evaluation by national standardization bodies. We keep our ear on the ground and intend to enrich our application *Geometrical Tolerancing* with new symbols and concepts when they will be accepted by the ISO members and new standard will be officially published by ISO.

REFERENCES

[1] Z. Humienny, "State of Art in Standardization in GPS area", CIRP Journal of Manufacturing Science and Technology, vol. 2 (2009), pp. 1–7.

[2] N. Anwer, A. Ballu, L. Mathieu, "The skin model, a comprehensive geometric model for engineering design", CIRP Annals – Manufacturing technology, v. 62, i. 1 (2013), pp. 143-146.

[3] H. S. Nielsen, "Recent developments in International Organization for Standardization geometrical product specification standards and strategic plans for future work", Proc. of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2013, vol. 227, no.5, pp. 643-649.

[4] T. Werner, A. Weckenmann, "Sustainable quality assurance by assuring competence of employees", Measurement vol. 45, i. 6 (2012), pp. 1534-1539.

[5] R. W. H Lau. N. Y. Yen, F Li, B. Wah, "Recent development in multimedia e-learning technologies", World Wide Web (2014) 17, pp. 189-198.

[6] T. Arndt, E. Katz, "Visual software tools for multimedia authoring", Journal of Visual Languages & Computing, vol. 21, i. 3, 2010, pp. 184-191.

[7] Z. Humienny, M. Berta, "A digital application for geometrical tolerancing concepts understanding", Procedia CIRP 27 (2015), p. 264 – 269.

[8] http://www.adobe.com/devnet/flash/articles /flash_key_concepts.html, access June 2015.