

SOME SPECIFIC PROCEDURES OF GEOMETRICAL MODELLING IN THE PrESMod LANGUAGE FOR 2D PARAMETER GEOMETRIC MODELLING

Željko Popović

Original scientific paper

The PrESMod modelling language uses lines/commands for the description of entities and modifications such that characteristic points, coordinates of characteristic points and other geometric parameters, relational position expressions, functional entity descriptions and also three types of relational statements for selecting one of two possible point sets can be used in the geometric description. In this work solutions have been analysed for problems of geometric modelling when based on one part of the modelling procedure, after specifying characteristic points, coordinates of characteristic points, characteristic geometric parameters, such as diameter, radius, angle, length and also parallel, normal, tangent relations, there are several solutions for the given entity (how to obtain one solution from two or more solutions). Finally, two commands in the PrESMod language for 2D parameter geometric modelling are given.

Keywords: 2D geometric modelling; parametric modelling; PrESMod; programming language

Neki specifični postupci geometrijskog modeliranja u jeziku PrESMod za 2D parametarsko geometrijsko modeliranje

Izvorni znanstveni članak

U jeziku za modeliranje PrESMod, za opis entiteta i modifikacija rabe se takve linije/naredbe da se u primarnom (geometrijskom) opisu mogu rabiti karakteristične točke, koordinate karakterističnih točaka i drugi geometrijski parametri, relacijski iskazi položaja, opis entiteta funkcijama, kao i tri tipa relacijskih iskaza za izdvajanje jednog od dva moguća skupa točaka. Ovdje se razmatra rešenje problema geometrijskog modeliranja, kada na osnovu jednog dijela postupka modeliranja, nakon specificiranja karakterističnih točaka, koordinata karakterističnih točaka, karakterističnih geometrijskih parametara kao što su promjer, polumjer, kut, dužina, kao i relacija paralelno, okomito, tangencijalno nastaje više rješenja za dani entitet (kako se iz dva ili više rješenja dobiva jedno rješenje). Na kraju se daje izgled dvije naredbe u jeziku PrESMod za 2D parametarsko geometrijsko modeliranje.

Ključne riječi: 2D geometrijsko modeliranje; parametarsko modeliranje; PrESMod; programski jezik

1 General on the modelling procedure

Each basic graphical entity is modelled with a set of operations, subcommands and similar. The collection of operations for geometric modelling depends on the entity type. There are several collections of operations for geometric modelling of the same entity. The modelling process includes specification of points, separate point coordinates, other characteristic geometric parameters, and also position relations such as parallel, normal, tangent. Introduction of relational statement in the procedure of geometric modelling when there are several solutions to one part of the procedure facilitates entity modelling or defining the required solution. Three types of relational statement can be used in the procedure of geometric modelling. The first type of relational statement is something larger than something or something smaller than something or similar. The second type of relational statement is something larger or something smaller or similar. The third type of relational statement is something closer to something or something farther than something or similar expressions something closest to something or farthest from something. Expressions with standard mathematical operations, standard mathematical functions, variables, constants, coordinates of characteristic points and other characteristic geometric entity parameters can be used for specifying values for any point (set of coordinates), point coordinates separately and other characteristic geometric parameters.

2 Structure and application of relational expressions

Type 1 relational statement are expressions using point coordinates and other characteristic geometric entity

parameters on the left and right side of the relation larger, smaller and all relations containing larger and smaller. The expression on the left and right side of the relation can be specified in a similar way as when specifying geometric entity characteristics (coordinates of characteristic points, characteristic geometric parameters). Geometric modelling enables specification of type two relational statements. This expression type has an expression (larger, smaller or similar) on one side of the relation, with nothing on the other side. This expression type can generally be specified in a similar way as type 1 relational statement. This is a relational statement of the type "slightly larger", "slightly smaller", "something is not smaller" and similar. This type of expression defines one of two possible states for "something".

These two types of relational statement are used for entity modelling, when based on the remaining part of the geometric modelling procedure there can be two solutions, i.e. two sets of points for a given entity. In this case this statement specifies which of the two sets of points is selected. This is solved by including coordinate points and other characteristic geometric parameters in the relational statement. For type 2 relational statement it is solved by the request that values of some coordinates of a characteristic point or value of some characteristic parameter are larger or smaller (for example x coordinate of some characteristic point larger, angle smaller, diameter larger or similar). Geometric modelling enables specification of one or more relational statements of type 1 or 2. Specification of one of these two types of relational statements enables elimination of one of two possible sets of points resulting from the remaining entity modelling procedure.

The third type of relational statement is also used for separation of one solution from a group of solutions for a

modelled entity. This statement reduces the number of solutions to one or a group of solutions formed from the remaining entity modelling procedure. This type (type 3) of relational statement uses relations closer, further, closest, farthest. The left and right sides of these relations contain characteristic entity points. Compared to type 1 or 2 relational statement where specifying one relational statement reduces the number of solutions (sets of points for a given entity) when there are more than two to half the number of solutions obtained based on the remaining entity modelling procedure, the type 3 relational statement (statement something closest to something or something farthest from something) can immediately separate only one solution (only one set of points).

3 Examples of applying relational expressions

Applying relational statement type 1 and 2

One example of using relational statement type 1 and 2 is shown for a definition in the process of modelling a straight line.

If the lexic, semantic and syntactic structure of the PrESMod modelling language is used [1] the straight line *i* (Fig. 1) in the modelling procedure can be defined (described) as:

$M=am1,am2;$
 $//(k);$
 $L=al;$

$xn>xm$ (obtained i.e. modelled straight line MN' using the type 1 relational statement where it is defined that the x coordinate of end point is larger than the x coordinate of the starting point – Fig. 1);

$M=am1,am2;$
 $//(k);$
 $L=al;$

$xn<xm$ (obtained i.e. modelled straight line MN'' using the type 1 relational statement where it is defined that the x coordinate of end point is smaller than the x coordinate of the starting point).

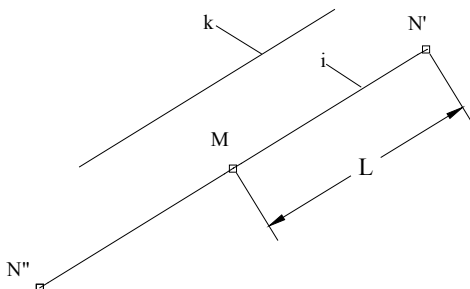


Figure 1 Straight line parallel to an already defined straight line

The type 2 relational statement for the straight line MN' could be $xn>$ where it is defined that the x coordinate of the end point is larger.

The type 2 relational statement for the straight line MN'' could be $xn<$ where it is defined that the x coordinate of the end point is smaller.

In this case:

M – the starting point; N – the end point; $//$ – parallel relation; L – straight line length

$xn>xm$ – relational statement; $xn<xm$ – relational statement

$xn>$ – relational statement; $xn<$ – relational statement; $am1, am2, al$ – expression, variable or constant.

Applying type 3 relational statement

When several solutions form as the result of input data such as characteristic points, coordinates of characteristic points, characteristic geometric parameters (diameter, radius, angle, length and similar) and relations (normal, parallel, tangent) then the modelling procedure uses relational statement containing relations farther, closer, farthest, closest. Characteristic points are included besides these relations in relational statement.

Examples:

a) If the lexic, semantic and syntactic structure of the PrESMod modelling language is used the circle *i* (Fig. 2) in the modelling procedure can be defined (described) as:

$IC(k);$
 $IC(l);$
 $R=ar$

and then four solutions are obtained.

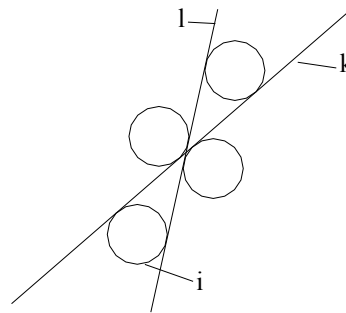


Figure 2 Circle tangent to two defined straight lines

The relational statement $C.closest.M (l)$ gives (separates) one solution i.e. the circle marked in Fig. 2 as *i*. (The straight lines tangent to the modelled circle are marked with *k* and *l*.)

In this case:

IC – tangent relation; R – circle radius; C – circle centre; $M(l)$ –starting point of entity *l*.

b) If the lexic, semantic and syntactic structure of the PrESMod modelling language is used the straight line given in Fig. 3 can be defined (described) in the modelling procedure as:

$M=am1,am2;$
 $IC(k);$
 $L=al$

and then four solutions are obtained.

The relational statement $N.closer.to.C(k)$ gives (separates) two solutions i.e. straight lines marked in Fig. 3 as *i'* and *l'*. (The circle tangent to the modelled straight line is marked as *k*.)

One more relational statement, for example $yn>$ to separates the desired straight line.

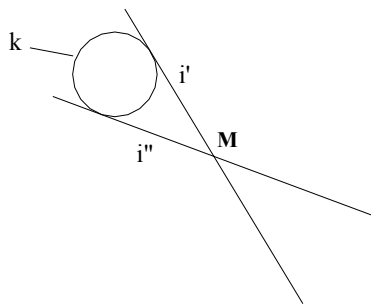


Figure 3 Straight line tangent to a defined circle

In this case:

M – the starting point; N – the end point ; IC – tangent relation; ; L – straight line length;

C(k) – circle centre k; $yn >$ - relational statement; am1, am2, al – expression, variable or constant.

Further are given definitions of entity with relational statements type 1, type 2 and type 3.

In Fig. 4 is shown an example of defining oriented line along with a starting point, with a length of the line, with x or y coordinate of endpoint and relational statement.

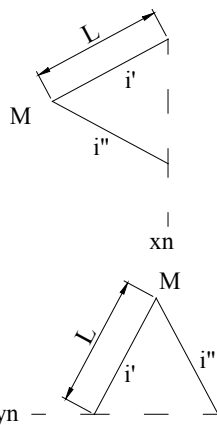


Figure 4 Straight line with a starting point, with a length of the line, with x or y coordinate of endpoint and relational statement

The general definition and examples of defining the line are given below.

General definition of a command (command line)

$i;L:M=am1,am2;xn=an1;L=al;$ relational statement (type 1, type2 or type3)

$I;L:M=am1,am2;yn=an2;L=al;$ relational statement (type 1, type 2 or type 3)

Examples of command lines for straight line modelling

- 13;L: M=20,25;xn=38;L=25;yn>
- 14;L: M=xm(5)+b,ym(5);xn=xm+c;L=d;yn<
- 15;L: M=xc(2),yc(2)+b;L=30;yn=35;xn<xm
- 16;L: M=20,50;yn=30;L=40;N.closer.specified point

If the difference between coordinates xm and xn i.e. ym and yn is equal to length L then it is not necessary to specify a relational statement for the position of the undefined coordinate yn i.e. xn.

In this case in the general command (command line) definition:

i – entity identification; L – denotation of the straight line entity; M – starting point; am1 and am2 – constants, variables or arithmetic expressions for setting values of x i.e. y coordinates of the starting point;

xn – x coordinate of the end point; an1 – constant, variable or arithmetic expression for setting values of the x coordinate of the end point;

yn – y coordinate of the end point; an2 – constant, variable or arithmetic expression for setting values of the y coordinate of the end point; L – straight line length; al – constant, variable or arithmetic expression for setting of the straight line length value. Three types of relational statement for separating one of two possible sets of points for the entity obtained based on the previous modelling procedure are given here.

Examples for straight line definition contain concrete relational statement mentioned in the examples given above.

In Fig. 5 is shown an example of defining circular arc with a starting point, with endpoint, with radius and relational statement type 1, type 2 or type 3.

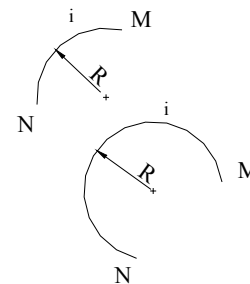


Figure 5 Circular arc defined with a starting point, endpoint, radius and relational statement

The general definitions and examples of defining circular arc are given below.

General definition of a command (command line)

$i;C:M=am1,am2;N=an1,an2;R=ar;$ relational statement (type 1, type 2 or type 3)

Examples of command lines for arc modelling

- 19;C: M=40,40;n=20,20;r=30
- 20;C: M=a,b;N=s(3);R=d5/2;yc>(ym+yn)/2
- 21;C: M=2,2;n=1,1;R=((xm-xn)^2+(ym+yn)^2)^.5/2.0
- 22;C: M=30,20;N=n(7);r=D4/2;L<
- 23;C:M=50,70;N=90,120;r=r(4);C|N(5)

If the centre is at the middle of the straight line joining points M and N ($MN=2*R$) then it is not necessary to give the relation for the arc centre position

In this case in the general command (command line) definition:

i – entity identification; C – denotation for the circular arc entity; M – starting point; am1 and am2 – constants, variables or arithmetic expressions for setting values of x i.e. y coordinates of the starting point; N – final point; an1 and an2 – constants, variables or arithmetic expressions for setting values of x i.e. y coordinates of the final point; R – radius of the circular arc entity; ar – constants, variables or arithmetic expressions for setting values of the radius of the circular arc.

One of three types of relational statement for separation of one of two possible sets of points for the circular arc entity obtained based on the previous modelling procedure is given here.

Examples for defining a circular arc entity contain concrete relational statement mentioned in the examples given above.

When describing entity 20 the relational statement was defined as somewhat wider $yc > (ym + yn) / 2$.

Meaning that the y coordinate of the centre of the circular arc is larger than the sum of the y coordinate of the starting point and the x coordinate of the end point divided by 2.

When describing entity 22 the length of the circular arc is smaller $L <$.

When describing entity 23 the centre of the circular arc is closer to the end entity point $5 C | N(5)$.

In Fig. 6 is shown an example of defining the circle with central point, with tangential relation and with relational statement type 2.

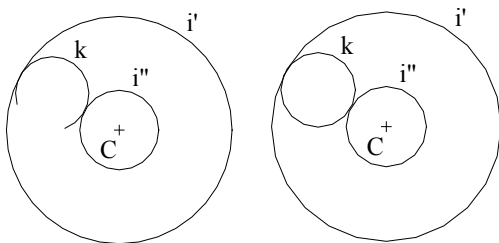


Figure 6 The circle defined with central point, tangential relation and relation statement type 2.

The general definition and examples of defining the circle are given below.

General definition of a command (command line)

$i;O: C=ac1,ac2;IC(k);$ relational statement Type 2 LT

Examples of command lines for circle modelling

- 11;O: C=25,30;IC(7);R>
- 12;O: C=45,50;IC(8);D<
- 13;O: C=xm(1)+b,70;R<;IC(9)

Here is, in the general definition, in the command line:

i – identification of the entity; O – symbol for the circle; C – symbol for the centre of the circle; $ac1,ac2$ – constant, variable or arithmetic expression; $IC(k)$ – indicates a tangential relation to the entity k .

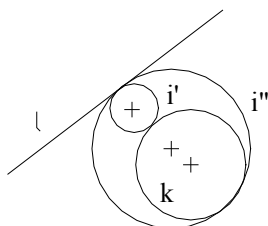


Figure 7 The circle defined with tangential relations on two entities and relational statement type 2

In the definition given in Fig. 6 is used relational statement type 2.

In Fig. 7 is shown an example of defining the circle with tangential relations and relational statement type 2.

The general definitions and examples of defining the circle are given below.

General definition of a command (command line)

$i;O: IC(k);IC(l);$ relational statement Type 2 LT

Examples of command lines for circle modelling

- 14;O: IC(3);IC(5);R>
- 15;O: IC(2);IC(4);D<

In the general definition, in the command line are:
 i – identification of the entity; O – symbol for the circle; $IC(k)$ – indicates a tangential relation to the entity k ; $IC(l)$ – indicates a tangential relation to the entity l .

In the definition given in Fig. 7 is used relational statement type 2.

In Fig. 8 is shown an example of defining the circle with tangential relations on two entities, radius and relational statement.

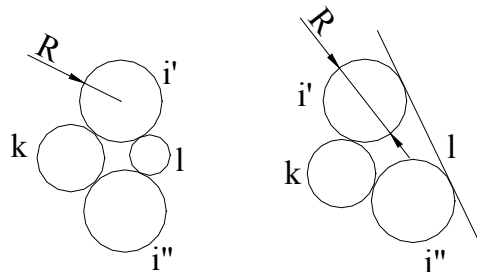


Figure 8 The circle is defined with tangential relations on two entities, with radius and with relational statement type 1, type 2 or type 3.

The general definitions and examples of defining the circle are given below.

General definition of a command (command line)

$i;O: IC(k);IC(l);R=ar;$ relational statement (type 1, type 2 or type 3)

Examples of command lines for circle modelling

- 11;L: IC(4);IC(7);R=15;yc>
- 12;L: IC(5);IC(8);R=a;yc<
- 13;L: IC(9);IC(2);R=R(11)+4;yc<yc(9)

In Fig. 8, in the general definition, in the command line are:

i – identification of the entity; O – symbol for the circle; $IC(k)$ – indicates a tangential relation to the entity k ; $IC(l)$ – indicates a tangential relation to the entity l ; R – symbol for the radius of the circle; ar – constant, variable or arithmetic expression.

In the definition given in Fig. 8 are used relational statements: type 1, type 2 or type 3.

4 Conclusion

The main advantage of the PrESMod modelling language is in the availability of all possible variations in entity description. Characteristic points, coordinates of characteristic points and characteristic parameters and also relational expressions used in the primary

(geometric) description enable a large number of variations in the description of each standard entity (straight line, circular arc, circle). Forms with concrete dimensions (constants), general dimensions (variables) and a combination of concrete and general dimensions can be described. In short, entity description in the PrESMod modelling language is suitable for the user as it is short, with great description diversity and high understanding of the description for the end user. The PrESMod 2D modelling language is easy to learn. Further work on this project requires completion of the development of the PrESMod modelling language and development of a pre-processor in such a way that when forms are described in the PrESMod modelling language this pre-processor or processor processes (with included concrete dimensions before processing or during processing) the form description (program) into one of the recognizable formats and gives a concrete illustration. The illustration is obtained with dimensions defined during the description, while variable (general) dimensions are either entered before processing the form description or are entered during form processing (if they are not selected or entered before processing the form description).

The PrESMod modelling language can be used for describing geometries of CNC machines, and general design.

Basic solutions on which the PrESMod modelling language is based are good for creating CAD or CAD/CAM packages.

Some entities could possibly be entered by speech using next generation computers that will be especially effective and does not change the essence of definitions in the PrESMod modelling language.

The German company Data M was interested in the project "PrESMod language for 2D geometric parametric modelling".

The Australian organization SCIE has put the paper from the project "PrESMod language for 2D geometric parametric modelling" on its list of international scientific publications (WSI).

Author's address

Mr Željko Popović, Lecturer
TEHNIKUM TAURUNUM,
Nade Dimić 4, 11080 Belgrade (Zemun), Serbia
E-mail: presmod@gmail.com

5 References

- [1] Popović, Z. Modelling Language PrESMod, monograph ISBN 86-902411-3-2, ASVCo, Belgrade, Serbia, (2001).
- [2] Putnik, G. D.; Popovic, Z.; Arandjelovic, I. PrESmod Language for 2D Geometric Modelling – Some Specific Definitions. // in Cagil G., Kubat C., Oztemel E, Taskin H (Eds.) Intelligent & Manufacturing Systems – Features, Strategies and Innovation (Proceedings of the 6th International Conference on Intelligent & Manufacturing Systems – IMS), Sakarya, Turkey, (2008).
- [3] Popović, Z.; Arandelović, I.; Nikolić, J. A Language for 2D Parametric and Geometric Modelling - PrESMod and Entity Selection. // Contemporary Engineering Sciences. 6, 5(2013), pp. 203-211. DOI: 10.12988/ces.2013.3316
- [4] Popovic, Z. The PrESMod Language for Modeling (2D). // in Proceedings of the 7th International Scientific and Expert Conference of the International TEAM Society, Belgrade, Serbia, (2015).