CAPP CUSTOMIZATION ON THE BASE OF OBJECT-ORIENTED APPROACH

Alexander Sharmazanashvili
CAD/CAM Division
Georgian Technical University
sharm@gtu.edu.ge

ABSTRACT

Implementation of Group Technology in manufacturing processes planning brings necessity in CAPP customization. System customization often called third party developing, provide opportunity to create and built in existing software own modules and establish custom software. However, development process of custom software is time consuming, difficult and requires highly skilled staff with good background in mathematics, programming and manufacturing technology. Paper describes an object-oriented approach of CAPP customization, which has been developed at Georgian Technical University and simplifies custom software development process.

KEYWORDS

CAPP, Group technology, typization, System architecture, Weight parameters, Object classes, Geometrical objects, Custom software.

1. PROBLEM

Group Technology has become an important technological innovation of Computer Aided Process Planning (CAPP). Several manufacturing companies have been widely applied this methodology in multi-product development (Jiang Wen Bing. et al., 1992) and small-batch production.

The core of Group Technology is typization. Typical decisions are made regarding to manufacturing process and it is carried out for each particular part family (Mitrofanov S.P., 1959). Typization is made on the different levels of process description. Especially, formal description of operation sequence, operation structure, including cutting instrument and fixtures, machining stocks, tool movement rules, cutting conditions, relative CNC subroutines are given. According to each level, the corresponding decision-making model is built and realized as custom software of CAPP. On the base of this software, for each part from given part family, process planning activity is carried out (Tsvetkov V.D., 1972). While, for different part family, typization of new manufacturing processes and formation of new custom software of CAPP is required, its cause necessity of CAPP customization for each case.

Paper below describes CAPP customization methodology for the manufacturing process typization level of the machining stock, tool movement rules, cutting conditions and CNC subroutines description. Usually, custom software development for given level of process typization is time consuming and difficult, while development of geometrical calculations algorithms with conjunction of unique manufacturing experience formalization is required. Programming activity therefore requires highly skilled CAPP user with good background in mathematics, programming and manufacturing technology.

2. CONSIDERATION OF CAPP ARCHITECTURES

In CAD/CAM division at Georgian Technical University investigations are made in order to simplify CAPP users programming activity (Sharmazanashvili A., 1998). Different CAPP architectures in this case where considered. For comparative analysis four weight parameters, which can measure software development difficultness, where separated:

- Mathematical difficultness
- Logical difficultness
- Manufacturing technology difficultness
- Basic computer skills.

2.1. Ordinary architecture

In ordinary architecture *two* main units can be separated – CAPP engine and custom software (Figure 1). CAPP engine provide basic functions of system - graphical interaction, feature recognition, interpretation, visualization and documentation.

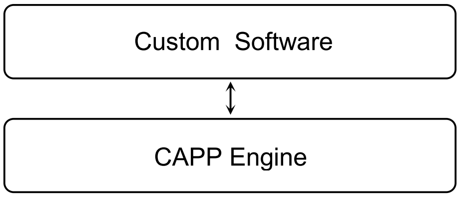


Figure 1. CAPP ordinary architecture

Custom software in this case provides system customization on part family and contains the full algorithm of typical manufacturing process realization. Therefore its development process is most difficult and the mentioned above parameters will be measured as 100% of difficultness.

2.2. Object oriented architecture

Different values will be received if object oriented approach were implemented. While custom software includes both, geometrical calculation algorithms and technological algorithms, it's preferable to separate *two* main classis of objects: geometrical and technological (Sharmazanashvili A., 1997). The purpose of geometrical object is calculation of original shapes from typical shapes presented in the parametric form

$$\mathbf{D:\Gamma\rightarrow D'} \qquad (1)$$

D = parametrical description of typical shape

D' = original description of considered part shape in the form of vector; shape is represented by a sequence of support point and numerical values of each point coordinates

Γ = transformation rules.

Usually, various types of shapes are associated with given part family. So, the class of geometrical objects is divided in several subclasses of objects and so on.

In same way, technological objects intended for calculation of tool path geometry and process condition parameters from corresponding typical descriptions

$$\mathbf{P:\Pi\rightarrow P'} \qquad (2)$$

$$\mathbf{C:\Omega\rightarrow C'} \qquad (3)$$

P,C = typical descriptions

P' = tool path geometry in the form of sequence of support points and its numerical coordinates in machine axis

C' = process condition parameters

Γ, Ω = transformation rules.

Different types of tool paths are usually separated and corresponding subclasses of objects are associated with them. Also, several process condition calculation rules with corresponding sub-classes are existed (Sharmazanashvili A., et al., 1998).

The mentioned above object classis unified as common part of custom software, called object system.

Another part of custom software is X-system, where the objects from different classes are linked in order to build process related algorithm and express typical manufacturing process originality. The main tasks of X-system programming are:

1) Selection of objects from object system
2) Provision of information linkage of objects through the set of assigns
3) Connection of X-system input parameters with parameters associated with objects.

The object-oriented CAPP architecture in this case contains the different configuration levels (Figure 2).

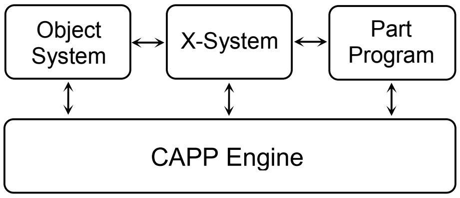


Figure 2. Object oriented architecture of CAPP

1st level unifies object system with CAPP engine and provide system general customization feature on the various types of manufacturing process – turning, milling, drilling, etc.

2nd level, unify X-system with object system and CAPP engine and provide system customization feature on various part families.

3rd level, unify part program with X-system, object system and CAPP engine and provide system customization feature on each part from part family.

2.3. Comparative analyze

Development of object system characterized with necessity of strong background in mathematical formalization and geometrical transformation methods. Also, deep knowledge of logic and programming methods of calculation steps, logical branches, cycles, parameters inspection procedures, etc. is required. Therefore the weight parameters value of mathematical and logical difficultness will be high and close with previous architecture case. Manufacturing technology difficultness parameter will be low, while object system is not express any typical manufacturing process and deep knowledge in manufacturing technology is not require. Special procedures of object programming, also increase requirements in basic computer skills.

Regarding to X-system, parameters of mathematical and logical difficultness will be low, because geometrical transformations are not require and only object parameters definition algorithms will be worked out.

High value of manufacturing technology difficultness parameter will be received. While X-system express the typical manufacturing process originality, strong background in manufacturing technology and process formalization methods is needed. Basic computer skills will be average.

Therefore, generation of solution and corresponding program for concrete part is much simplified. Mathematical and logical contribution is very low; also low is manufacturing technology programming difficultness, while process description requires only definition of parametrical modules from X-system.

Comparatively high level of basic computer skills is required to interact with X-system and CAPP engine.

Figure 3, below represents results of mentioned above analyzes. As it is shown, the summarized difficultness of custom software development is cutting down from object system to part program.

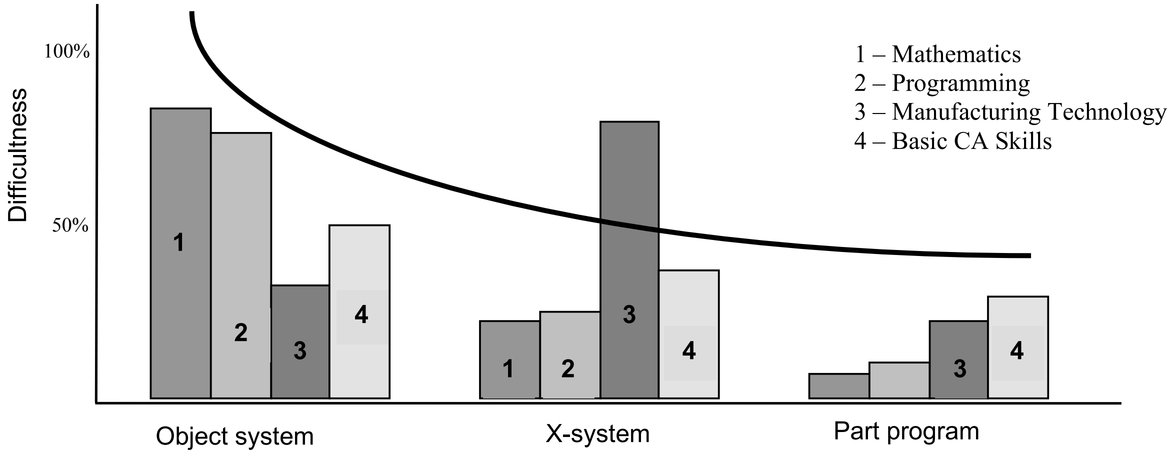
Advantage of suggested object oriented architecture is that the most difficult part of custom software, object system, is working out only one time, when the CAPP general customization on the manufacturing process types is made. While in ordinary architecture this is necessary to be done every time when manufacturing process typization is required.

Table 1

Description	*Skills*	*Background*	*Customization*
Object system	High	• Mathematics • Programming	Manufacturing process type
X-system	Med.	• Technology	Family of parts
Part program	Low	• Operator	Part

3. OBJECT SYSTEM DESCRIPTION

In CAD/CAM division at Georgian Technical University on the base of suggested approach object system for turning and X-system for FLANGE part family, was built. 10 main geometrical classis, half-open cylindrical, open



cylindrical, closed cylindrical, open face, closed face, half-open grooving, open grooving, closed grooving, cylindrical conjunction and grooving conjunction were separated. 142 objects of geometrical transformations were worked out for given classis of object. They permit to describe up to 95% of turning part surfaces. In special cases, where descriptions of non-formalized shapes are required, additional objects have to be worked out.

Three main features characterize each class of geometrical object:

- Structure, describes set of shapes the object geometry is consist of
- Topology, describes how typical shapes in object geometry is connected
- Parameterization, describes array of parameters, necessary for object geometry formal representation.

Figure 4, describes half open cylindrical class of objects.

Class : HALF_OPEN_CYLINDRICAL_STAIR
Objects : {G1, G2, G3, G4, G5, G6, G7}

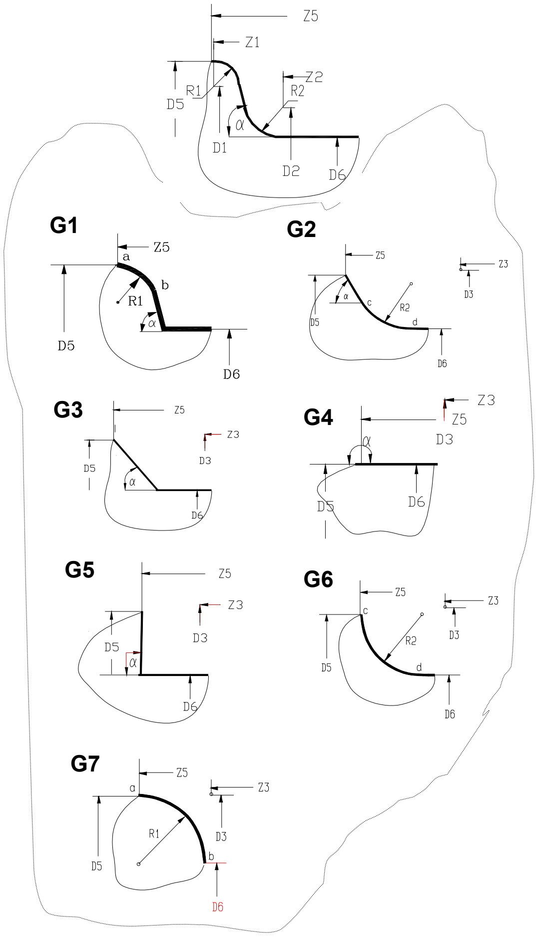


Figure 4. Half open cylindrical class of objects

This class is intended for typical shapes formed by tangential conjunction of two arcs with conical line and has topology "Arc-Line-Arc-Line". Parameterization describes 12 geometrical parameters:

Class *'Arc-Line-Arc-Line'*:
{D5,Z5,D6,D3,Z3,ALF,R1,R2,D1,Z1,D2,Z2}
(4)

The original set of values of parameters are corresponding to each object from considered class. Among this set, in each case, the key parameters can be identified, which describe condition of object separation. For example zero value of parameter *R2* describes condition for separation of G1 object.

Table 2

Class	Parameters		
	Required	Key	Additional
G1:'Arc-Line-Line'	D5,Z5,D6, ALF,D3,Z3, R1	R2=0	D1,Z1,D2, Z2

If *R1* equals to zero, another object G2 'Line-Arc-Line' will be received

Table 3

Class	Parameters		
	Required	Key	Additional
G2:'Line-Arc-Line'	D5,Z5,D6, ALF,D3,Z3, R2	R1=0	D1,Z1,D2, Z2

In case, if both *R1* and *R2* parameters equal to zero, according to value of *ALF* parameter, the following objects will be separated

Table 4

Class	Parameters		
	Required	Key	Additional
G3:'Line-Line'	D5,Z5,D6, D3,Z3	R1=0 R2=0 0<ALF<90	D1,Z1,D2, Z2
G4:'Line'	D5,Z5,D6, D3,Z3	R1=0 R2=0 D5=D6 ALF=180	D1,Z1,D2, Z2
G5:'Line-Line'	D5,Z5,D6, D3,Z3	R1=0 R2=0 ALF=90	D1,Z1,D2, Z2

If *ALF* equal to zero, whether *R1* equal to zero, or *R2* equal to zero, defines conditions for separation of G6 or G7 objects.

Table 5

Class	Parameters		
	Required	Key	Additional
G6:'Arc'	D5,Z5,D6, D3,R2	R1=0 ALF=0	D2, Z2, Z3
G7:'Arc'	D5,Z5,D6, D3,R1	R2=0 ALF=0	D1, Z1, Z3

Values of additional parameters identify conditions for separation of sub-classis from considered above object class.

In case, when *D1,Z1,D2,Z2* parameters value are not equal to zero, separation of sub-objects with topology of non-tangential conjunctions of arc and line is carried out. Table 6 describes all sub-objects to be received.

Table 6

Class	Sub-class	Parameters		
		Required	Key	Additional
G6	G6-1:'Arc'	D5,Z5,D6, D3,Z3,R2, Z2,D2	R1=0 0<ALF ≤90	D1,Z1
G6	G6-2:'Line-Arc'	D5,Z5,D6, D3,Z3,R2, Z2,D2	R1=0 ALF=0	D1,Z1
G7	G7-1:'Arc'	D5,Z5,D6, D3,Z3,R1, Z1,D1	R2=0 ALF=0	D2,Z2
G7	G7-2:'Arc-Line'	D5,Z5,D6, D3,Z3,R1, Z1,D1	R2=0 0<ALF ≤90	D2,Z2
G1	G1-1:'Arc-Line-Line'	D5,Z5,D6, D3,Z3,R1, ALF	R1=0 D1=0 Z1=0 Z2≠0	D2
G2	G2-1:'Line-Arc-Line'	D5,Z5,D6, D3,Z3,R2, ALF	R1=0 D2=0 Z2≠0	D1,Z1
G0	G0-1:'Arc-Line-Arc-Line'	D5,Z5,D6, D3,Z3,R2, R1,ALF	Z1=0 D1=0 D2=0 Z2≠0	-

Particular attention in this case is played to *Z2* parameter, while non zero value of *Z2* with different combination of non zero values of other parameters – *Z1,ALF,R1,R2* identify conditions for separation sub-objects with non-tangential conjunctions of arc with internal line. In rest of cases non-tangential conjunctions of arc are made with external lines (lines from other objects). Corresponding shapes are presented on figure 5.

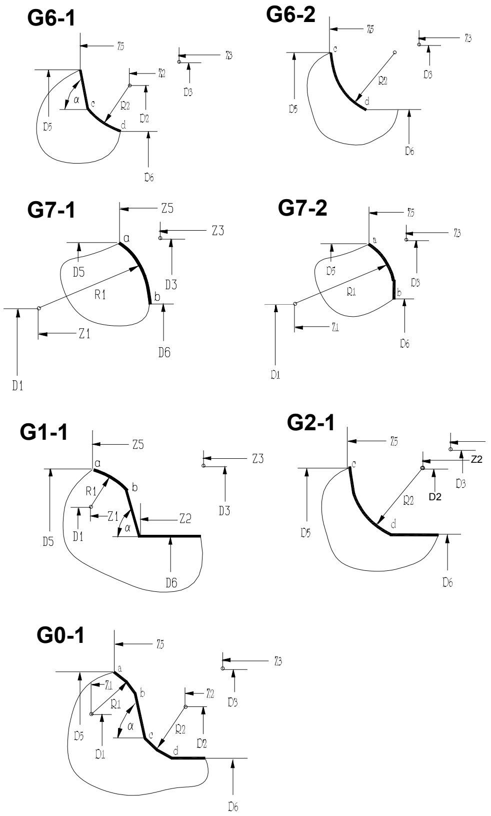


Figure 5. Object sub-classes

Mathematical model of each object contains set of equation, which expresses functional dependence of shapes support points on object parameters. According to parameterization, each support point is described by different set of parameters and for its calculation corresponding equation have to be separated.

Considered class of half open cylindrical objects contains five support points T1-T2-T3-T4-T5 (Figure 6).

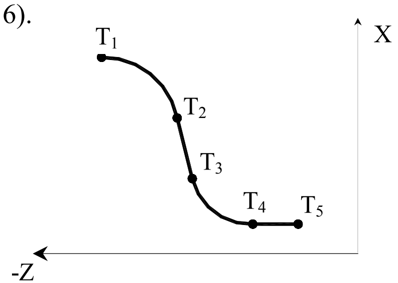


Figure 6

T_1 is described directly by parameters {D5,Z5} from (4)

$$\left\{ \begin{array}{l} x_1 = \frac{D5}{2} \\ z_1 = Z5 \end{array} \right\} \qquad (5)$$

T_2 is described by {D5, Z5, R1, α}

$$\left\{ \begin{array}{l} x_2 = \frac{D5}{2} - R1 \cdot (1 - \cos\alpha) \\ z_2 = Z5 - R1 \cdot \sin\alpha \end{array} \right\} \qquad (6)$$

T_3 is described by {D5, Z5, D6, R1, R2, α}

$$\left\{ \begin{array}{l} x_3 = \frac{D6}{2} - R2 \cdot (1 - \cos\alpha) \\ z_3 = Z5 - R1 \cdot \sin\alpha + ctg\alpha \cdot [\frac{D5 - D6}{2} - \\ - (R1 + R2) \cdot (1 - \cos\alpha)] \end{array} \right\}$$

(7)

T_4 is described by {D5, Z5, D6, R1, R2, α}

$$\left\{ \begin{array}{l} x_4 = \frac{D6}{2} \\ z_4 = Z5 - \sin\alpha \cdot (R1 - R2) + ctg\alpha \cdot [\frac{D5 - D6}{2} - \\ - (R1 + R2) \cdot (1 - \cos\alpha)] \end{array} \right\}$$

(8)

T_5 is described by {D6, Z3}

$$\left\{ \begin{array}{l} x_5 = \frac{D6}{2} \\ z_5 = Z3 \end{array} \right\} \qquad (9)$$

In case when the D1, Z1, D2, Z2 is not equal to zero, non-tangential conjunctions of arc and line in T_1 and T_4 are carried out. So, in (6), (7), (8), D5 replaced with D1 and D6 with D2.

The original set of equation has to be received from described general set for each object.

G01 According to table 2, T_3=T_4. Therefore,

$$\left\{ \begin{array}{l} x_1 = \frac{D5}{2} \\ x_2 = \frac{D5}{2} - R1 \cdot (1 - \cos\alpha) \\ x_3 = x_4 = x_5 = \frac{D6}{2} \\ z_1 = Z5 \\ z_2 = Z5 - R1 \cdot \sin\alpha \\ z_3 = z_4 = Z5 - R1 \cdot \sin\alpha + ctg\alpha \cdot [\frac{D5 - D6}{2} - \\ - R1 \cdot (1 - \cos\alpha)] \\ z_5 = Z3 \end{array} \right\} \qquad (10)$$

G02 According to table 3, T_1=T_2. Therefore,

$$\left\{ \begin{array}{l} x_1 = x_2 = \frac{D5}{2} \\ x_3 = \frac{D6}{2} + R2 \cdot (1 - \cos\alpha) \\ x_4 = x_5 = \frac{D6}{2} \\ z_1 = z_2 = Z5 \\ z_3 = Z5 + ctg\alpha \cdot [\frac{D5 - D6}{2} - R2 \cdot (1 - \cos\alpha)] \\ z_4 = Z5 - R2 \cdot \sin\alpha + ctg\alpha \cdot [\frac{D5 - D6}{2} - R2 \cdot (1 - \cos\alpha)] \\ z_5 = Z3 \end{array} \right\}$$

G03 According to table 4, T_1=T_2 and T_3=T_4. Therefore,

$$\left\{ \begin{array}{l} x_1 = x_2 = \frac{D5}{2} \\ x_3 = x_4 = x_5 = \frac{D6}{2} \\ z_1 = z_2 = Z5 \\ z_3 = z_4 = Z5 - ctg\alpha \cdot (\frac{D5 - D6}{2}) \\ z_5 = Z3 \end{array} \right\} \qquad (11)$$

G04 According to table 4, T_1=T_2 and T_3=T_4=T_5. Therefore,

$$\left\{ \begin{array}{l} x_1 = x_2 = x_3 = x_4 = x_5 = \frac{D5}{2} \\ z_1 = z_2 = Z5 \\ z_3 = z_4 = z_5 = Z3 \end{array} \right\} \qquad (12)$$

G05 According to table 4, T$_1$=T$_2$ and T$_3$=T$_4$. Therefore,

$$\left\{\begin{array}{l} x_1 = x_2 = \dfrac{D5}{2} \\ x_3 = x_4 = x_5 = \dfrac{D6}{2} \\ z_1 = z_2 = z_3 = z_4 = Z5 \\ z_5 = Z3 \end{array}\right\} \tag{13}$$

G06 According to table 5, T$_1$=T$_2$=T$_3$ and T$_4$=T$_5$. Therefore,

$$\left\{\begin{array}{l} x_1 = x_2 = x_3 = \dfrac{D5}{2} \\ x_4 = x_5 = \dfrac{D6}{2} \\ z_1 = z_2 = z_3 = Z5 \\ z_4 = z_5 = Z5 - R2 \end{array}\right\} \tag{14}$$

G07 According to table 5, T$_2$=T$_3$=T$_4$=T$_5$. Therefore,

$$\left\{\begin{array}{l} x_1 = \dfrac{D5}{2} \\ x_2 = x_3 = x_4 = x_5 = \dfrac{D6}{2} \\ z_1 = Z5 \\ z_2 = z_3 = z_4 = z_5 = Z5 - R1 \end{array}\right\} \tag{15}$$

For Sub-classis of objects, according to table 6, following models are separated:

G6-1 T$_1$=T$_2$ and T$_4$=T$_5$

$$\left\{\begin{array}{l} x_1 = x_2 = \dfrac{D5}{2} \\ x_3 = \dfrac{D2}{2} - R2 \cdot \cos\alpha \\ x_4 = x_5 = \dfrac{D6}{2} \\ z_1 = z_2 = Z5 \\ z_3 = Z5 - ctg\alpha \cdot (\dfrac{D5 - D2}{2} + R2 \cdot \cos\alpha) \\ z_4 = z_5 = Z3 \end{array}\right\} \tag{16}$$

G6-2 T$_1$=T$_2$=T$_3$ and T$_4$ =T$_5$

$$\left\{\begin{array}{l} x_1 = x_2 = x_3 = \dfrac{D5}{2} \\ x_4 = x_5 = \dfrac{D6}{2} \\ z_1 = z_2 = z_3 = Z5 \\ z_4 = z_5 = Z3 \end{array}\right\} \tag{17}$$

G7-1 T$_2$=T$_3$=T$_4$=T$_5$

$$\left\{\begin{array}{l} x_1 = \dfrac{D5}{2} \\ x_2 = x_3 = x_4 = x_5 = \dfrac{D6}{2} \\ z_1 = Z5 \\ z_2 = z_3 = z_4 = z_5 = Z3 \end{array}\right\} \tag{18}$$

G7-2 T$_3$=T$_4$=T$_5$

$$\left\{\begin{array}{l} x_1 = \dfrac{D5}{2} \\ x_2 = \dfrac{D1}{2} + R1 \cdot \cos\alpha \\ x_3 = x_4 = x_5 = \dfrac{D6}{2} \\ z_1 = Z5 \\ z_2 = Z1 - R1 \cdot \sin\alpha \\ z_3 = z_4 = z_5 = Z3 \end{array}\right\} \tag{19}$$

G1-1 T$_3$=T$_4$

$$\left\{\begin{array}{l} x_1 = \dfrac{D5}{2} \\ x_2 = \dfrac{D6}{2} + \dfrac{Z2 \cdot (\dfrac{D1 - D6}{2} + R1 \cdot \cos\alpha)}{Z1 - R1 \cdot \cos\alpha - ctg\alpha \cdot (\dfrac{D1 - D6}{2} + R1 \cdot \cos\alpha)} \\ x_3 = x_4 = x_5 = \dfrac{D6}{2} \\ z_1 = Z5 \\ z_2 = Z2 + \dfrac{Z2 \cdot ctg\alpha \cdot (\dfrac{D1 - D6}{2} + R1 \cdot \cos\alpha)}{Z1 - R1 \cdot \sin\alpha - ctg\alpha \cdot (\dfrac{D1 - D6}{2} + R1 \cdot \cos\alpha)} \\ z_3 = z_4 = Z2 \\ z_5 = Z3 \end{array}\right\} \tag{20}$$

$\boxed{\text{G2-1}}$ $T_1=T_2$

$$\left\{\begin{array}{l} x_1 = x_2 = \dfrac{D5}{2} \\ x_3 = \dfrac{D6}{2} + R2\cdot(1-\cos\alpha) \\ x_4 = x_5 = \dfrac{D6}{2} \\ z_1 = z_2 = Z5 \\ z_3 = Z5 - ctg\alpha\cdot[\dfrac{D5-D6}{2} - R2\cdot(1-\cos\alpha] \\ z_4 = Z2 + \sqrt{R_2^2 - (\dfrac{D2-D6}{2})^2} \\ z_5 = Z3 \end{array}\right\}$$

(21)

$\boxed{\text{G0-1}}$ $T_1=T_2$

$$\left\{\begin{array}{l} x_1 = \dfrac{D5}{2} \\ x_2 = \dfrac{D1}{2} + R1\cdot\cos\alpha \\ x_3 = \dfrac{D2}{2} - R2\cdot\cos\alpha \\ x_4 = x_5 = \dfrac{D6}{2} \\ z_1 = Z5 \\ z_2 = Z1 - R1\cdot\sin\alpha \\ z_3 = Z1 - R1\cdot\sin\alpha + ctg\alpha\cdot[\dfrac{D1-D2}{2} - (R1+ \\ +R2)\cdot(1-\cos\alpha] \\ z_4 = Z2 + \sqrt{R_2^2 - (\dfrac{D2-D6}{2})^2} \\ z_5 = Z3 \end{array}\right\}$$

(22)

Same objects and corresponding models are also built for the other classis.

Tool movement objects permit to describe the support points of tool path according to part shape separated by geometrical objects and current position of tool.

Three main classis of tool movement objects were separated for considered Object system:

M1 – 4 point closed cycle movement

M2 – 3 point closed cycle movement

M3 – Equidistant movement.

M1 objects provide calculation of P_2 conducted point from the P_1 starting point across the X or Z axis parallel line; also, P_3 and P_4 points are calculated to provide tool back movement. So, each object of given class is characterized by 3 typical movement. Depended on weather, this movement are carried out fast, or on federate, *two* different sub-classis of M1 objects are separated, with topology:

M1-1 –“Federate->Fast->Fast”

M1-2 – “Federate->Federate->Fast”

M1-1 class of objects provide tool movement on federate from P_1 starting point up to P_2 conjunct point, which is placed on part surface (Figure 7);

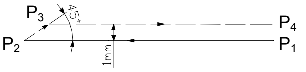


Figure 7

then fast movement across the 45^0 angled line up to P_3 point with transferring on 1mm and back fast movement in P_4 point. 8 different objects are separated from given sub-class according to left-right,-up-down directions of P_1, P_2 movement and P_2, P_3 fast movement.

M1-2 class of objects provide tool movement on federate from P_1 starting point to P_2 point (Figure 8); then federate movement up to P_3 point across the part surface with transferring on predefined depth of cut (*t*) and back fast movement in P_4 point is carried out.

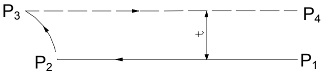


Figure 8

Also, 8 different objects are separated from M1-2 class, according to left-right-up-down directions of P_1, P_2 and P_2, P_3 federate movement.

M2 objects provide calculation of P_2 conjunct point from the P_1 starting point across the X or Z axis parallel line. Tool movement is starting on federate from P_1 point up to P_2 point and finished by back fast movement in P_1 point (Figure 9). So,

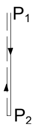


Figure 9

all objects from given class have the same topology – “Fast-Fast”. According to left-right-up-down directions of P_1, P_2 movement, 4 different objects were separated.

M3 objects provide calculation of points for equidistant movement. Number of points is depended on the part shape, which is “copied” during equidistant movement and is limited by geometrical objects. There are two sub-classis of M3 objects:

M3-1 provide equidistant movement with scaling (Figure 10) and M3-2, equidistant movement without scaling (Figure 11). According to left-right-up-down directions of movement, 4 different objects from each sub-class were separated.

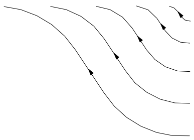


Figure 10

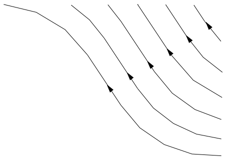


Figure 11

Finally, 28 different tool movement objects were built.

Objects for optimization of process condition parameters were worked out from general representation of optimization model (Sharmazanashvili at Al., 1988). According to this representation optimal values of process parameters can be found by simultaneous solving of *two* equations of boundary conditions

$$\left\{ \begin{matrix} H = C_H \cdot V^{\alpha_H} \cdot S^{\beta_H} \cdot t^{\gamma_H} \\ \Phi = C_\Phi \cdot V^{\alpha_\Phi} \cdot S^{\beta_\Phi} \cdot t^{\gamma_\Phi} \end{matrix} \right\} \tag{23}$$

Gornev V.F. (1980) suggests to joint in the [H] class, equations that are relatively substantially depended on federate (S) and in the [Φ] class, equations that are rather depended on cutting speed (V). By result of analyze of boundary conditions described in mentioned above source, *five* different objects of optimization were separated and provide optimization:

[PV] – by restrictions of P=*Const* and V=*Const*, where, P - cutting force and V - cutting speed.

[ST] – by restrictions of S=*Const* and T=*Const*, where, S - federate and T - tool life period.

[SN] – by restrictions of S=*Const* and N=*Const*, where, N – cutting power.

[MN] – by restrictions of M=*Const* and N=*Const*, where, M – cutting moment.

[MT] – by restrictions of M=*Const* and T=*Const*.

Object system software was built on the APT similar language for object-oriented engine Turbo T. System is placed into the text files where source code is presented. Turbo T engine carry out interpretation of source code and generation of solution in the standard CLDATA form. Language possibilities are described in Table 7.

Table 7

Formal Parameters	A, B, C, A1, Z5, DEPTH, ...
Functions	SIN, COS, TAN, ATAN, ASIN, SQRT, ROUND, SQR
Mathematical expressions	A=SQRT(B) + SIN (ROUND (D*2/B)) / (C-1)^2
Labels	<Numbers>, <Strings>
Tool Axis motion operator	X <value>, Z <value>
Conditional movement operator	IF (<value>) <value>
Unconditional movement operator	Goto <value>
Cycle and assignment operators	Macro (<value>)
Cutting conditions defining operators	Federate (<value>) / Fast
Tool fix operator	Tool (<value>)

CONCLUSIONS

(a) Separation of *two* main parts in custom software brings advantage of suggested object oriented architecture of CAPP system. While it brings possibility to move most difficult part of custom software, related with geometrical transformations, in to the lowest level of system architecture and carry out its development process only one time during system customization.

(b) Manufacturing process-related algorithms are worked out on higher level of system architecture. Corresponding custom software development process is carried out each time when process typization for group technology is necessary and it's requires users with strong knowledge only in manufacturing technology, without any strong background in mathematics and programming.

(c) Development of object system for turning at Georgian Technical University, shown that the corresponding tasks are well formalized and language with pure programming ability can be implemented.

(d) For turning objects system 142 geometrical, 28 tool movement and 5 optimization objects were built. They can easily share between different part families and permits to cover up to 95% of cases.

(e) Suggested approach can be adopted on other types of machining operations without any considerable changes.

REFERENCES

Jiang Wen Bing, Wang Zhi Bo, Sun Hou Fang (1992) "Group Technology Application in China"; *in Technovation*, vol:12, Iss:8, pp.509-514

Mitrovanov S.P. (1959) "The Fundamentals of Group Technology"; *Leningrad*, p.265 (in Russian)

Tsvetkov V.D. (1972) "Implementation of Computers for Designing of Technological Process of Machining"; *Mashinostroenie,* Moscow, p.237 (in Russian)

Gornev V.F., (1989), "Integration of Technological operation control systems"; *Proceedings of first all union meeting of manufacturers*, Moscow, pp.41-43. (in Russian)

Sharmazanashvili A. (1998) "The State-of-the-art in trends of CAD/CAM Development"; *in Transactions of Georgian Technical University,* no.1 (417), pp.100-110

Sharmazanashvili A., Megrelishvili L (1998) "Feature-based approach in CAD/CAM/CNC Integration"; *in Transactions of 2^{nd} International Symposium of Tools and Methods of Concurrent Engineering, 1998, 19-21 April, Manchester, UK,* pp.423-431

Sharmazanashvili A., Kharadze I. (1998) "Implementation of Heuristic Models for the Parametric Optimization of Turning"; *Proceedings of the ParametricCAD'98 workshop*, no.3, GTU,pp.54-62

Sharmazanashvili A. (1997) "Consideration of Para/CAM – the Object Oriented Process Planning Software"; *Proceedings of the ParametricCAD'97 workshop,* no.2, GTU, pp.7-15