Computer-aided Design for Robotic Assembly Technology

Igor Sh. Nevlyudov, Olexander M. Tsymbal, Svetlana S. Milyutina

Abstract — **There is proposed the logical model of assembly technological processes design. The basic set of predicates, which characterizes the assembly process, is considered. The model is implemented by Prolog language program. The models of technological process design, based on semantic nets and on frames and a method of voice information input are proposed. Also the questions of decision-making for robot manipulations are considered.**

Index Terms **— assembly, logics, predicate, robotics.**

I. Introduction

The radio electronic devices technological processes design is a field, where the role of the human factor is still great even for simple products.

Generally, the assembly technology has descriptive character and does not allow computing methods using because of the following features [1]:

- Absence of analytical dependences;
- Complex interrelation and mutual influence of different problems;
- A great role of empirical dependences and existence of implicit objective laws;
- Huge information volumes and opposite factors.

Such prominent features of technology and its ill-structured character attract the attention of specialists in field of knowledge presentation and artificial intelligence methods.

During the technological processes models design it is necessary to take in account the huge experience, which has been collected by technologists, usually in unstructured view. In the same time it is necessary to start from the presence of quite determined informal laws of technology design.

Robots as programmable automatic devices can be used on such operations as equipment service, painting and control. The assembly operations are the most perspective direction of industrial robots application. The creation of assembly operations models allows simplifying of assembly technological processes design, especially if representation of model and following robot decision-making system design will be on the common mathematical and logical basis [2].

The development of robotic decision-making support system contains the significant volume of logic programming [3]. So it looks very actual and perspective to include the elements of decision-making system to the computer-aided system for technological assembly design.

II. Assembly Process and Technological Rules

The assembly technological process is sequence of technological operations on junction of parts, which is ordered as to technical and economical requirements.

As to such definition, the assembly technological process has the functional links of two kinds:

- between the joints and parts of manufacturing system;
- the order of joint execution.

The basis requirements to assembly order can be expressed by following statements:

- if part is a technological base for the other part as to conditions of access and basing, then base would be set earlier then the other part;
- if part is restricted on access by other part, the restricted part would be set earlier, then the restricting one.

The process of assembly order can be presented in the following view [1]:

- disintegration of object into technological assembly units;
- definition of order on assembly units installation;
- definition of order on assembly units juncture.

The following sections will consider the models assembly technological processes automated design.

III. Logical Model

As it was said, technology is a science, built on a number of informal laws, while based on designer logic. The last one defines how the separated parts will be mounted to the single devise. So, the first model is logical.

Let's consider the setting of attached elements on the board of electronic device. It's need to set on the board the microchips, with the number between 0 and some i, condensers ($0 \div j$), transistors ($0 \div k$), resistors ($0 \div l$), inductors ($0 \div m$), diodes ($0 \div n$).

The description of given facts in the predicates logic may be represented as a next:

Manuscript received June 7, 2009.

Igor Sh. Nevlyudov is with the Kharkov National University of Radio Electronics. Address: Ukraine, 61166, Kharkov, Lenin Ave., 14, phone +38(057)7021486, e-mail: tapr@kture.kharkov.ua.

Olexander M. Tsymbal is with Kharkov National University of Radio Electronics. Address: Ukraine, 61166, Kharkov, Lenin Ave., 14, phone +38(057)7021486, e-mail: mcdulcimer@ukr.net.

Svetlana S. Milyutina is with the Kharkov National University of Radio Electronics. Address: Ukraine, 61166, Kharkov, Lenin Ave., 14, phone +38(057)7021486, e-mail: tapr@kture.kharkov.ua.

```
detail (board)
type (board, board)
detail (microchip_1)
type (“microchip_1”, microchip)
……………………
detail (microchip_i)
type (microchip_i, microchip)
detail (condenser_1)
type (“condenser_1”, condenser)
……………………
detail (condenser_j)
type (condenser_j, condenser)
detail (transistor_1)
type (“transistor_1”, transistor)
……………………
detail (transistor_k)
type (transistor_k, transistor)
detail (resistor_1)
type (resistor_1, resistor)
……………………..
detail (resistor_l)
type (resistor_l, resistor_l)
detail (inductor_1)
type (inductor_1, inductor)
……………………
detail (inductor_m)
type (inductor_m, inductor)
detail (diode_1)
type (diode_1, diode)
……………………
detail (diode_n)
type (diode_n, diode)
```

Every detail is characterized by parameters set:

parameters (name, square, size, setup_version). (1)

Setup version includes which leads element has (planar or core). For radio electronic apparatus assembly this feature may be written in next way:

peculiarity (name, [control_points_list]), (2)

where [control_points_list]=[cp1, cp2],
where cp1,cp2 – element control points.
Then assembly unit may be represented in next way

assembly_unit (number, board, [attached_elements list]) (3)

For radio electronic devices assembly there can be written the the next expression can be written

$$\begin{aligned}&\exists x\exists y(\mathrm{type}(x,\mathrm{board})\wedge \mathrm{type}(y,[\mathrm{attached\_elements\_list}])\\&\rightarrow \exists u(\mathrm{type}(u,\mathrm{assembley\_unit})\wedge \mathrm{include}(u,[x,y])).\end{aligned} \tag{4}$$

The product is considered as assembled when the entire elements are assembled. The assembly unit is assembled, when all details it consists are found.

$$\begin{aligned}&\exists B\exists D\exists u(\mathrm{base\_found}(B)\wedge \mathrm{compatible\_}\det \mathrm{ail}(B,D)\wedge\\&\mathrm{assembly\_unit\_found}(u,B,D)\wedge\\&\mathrm{assembled}(u))\rightarrow \mathrm{process\_executed}.\end{aligned} \tag{5}$$

In this way, at the first step the base detail has to be found. Next, detail connected to base is looked for and assembly unit, which includes the base and connected details, has to be found.

Mostly, the base detail has the biggest size:

$$\begin{aligned}&\exists B(\mathrm{base\_found}(B)\vee \mathrm{max\_size}(B))\\&\rightarrow \mathrm{base\_found}(B).\end{aligned}\ . \tag{6}$$

The detail, which has to be connected to the base, may have the necessary properties:

$$\begin{aligned}&\exists B\exists D\exists S(\det \mathrm{ail}(B)\wedge \det \mathrm{ail}(D)\wedge \mathrm{peculiarity}(B,S)\wedge\\&\mathrm{peculiarity}(D,S))\rightarrow \mathrm{compatible\_}\det \mathrm{ail}(B,D).\end{aligned}\ . \tag{7}$$

Then, it’s need to determine the order of assembly units assembling and the details, will connect base and compatible details.

The developed model is the base for program on Turbo-Prolog. The main predicate defines the setup of assembly order and has the next view:

```
make_process: –  find_base(Base),
   find_compatible(Base,D), find_joiner(Base,D,L),
   find_joint(Base,D,L,UL), assemble_joint(UL),
   make_process.
```

Thus, at first step the basis part Base is determined. Further the part D compatible with Base must be found, also part L, which is a connector for basis and its compatible part. On the following step the assembly unit’s list UL, which provides the connection of base and compatible details, must be found. The fulfillment of joints for the assembly unit’s list will mean the execution of whole assembly process, which is repeated in recursion.

So, the logical model of assembly technological process for radio electronic apparatus example is proposed.

IV. Semantic Net Model

Semantic models usually define the reasonable connections of objects. Every assembly units unites a number elements, logically (as it’s shown above) and reasonably connected. So now let’s consider the model of assembly technological process for the same radio electronic apparatus, presented in the previous part, using semantic nets.

Semantic net model assumes that there is the set of vertexes, which presents the object of simulation (and it’s elements), and the set of arcs, that describes the relations between objects. Thus, we have to set a group of attached elements to the board and to describe their relationship. Fig.1 presents it.

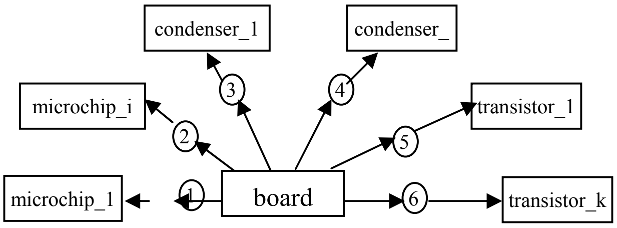


Fig. 1. The relation "Orientation"

V. Framework Model

On assembly technological processes design, there can be the standard sequences of unit assembling, what's why seem reasonable to consider the framework model of assembly process.

For any assembly product (AP) there can be defined a number of possible assembly schemes (AS) may exist. So AP=<AS(D)$_1$, AS(D)$_2$,…, AS(D)$_N$> – the set of possible assembly schemes, N – the number of possible assembly process schemes. AS(D) may be written in the next way:

$$AS(D)=<d_a, d_b,\dots,d_m>, \qquad (8)$$

where D – the set of necessary assembly operations,

$d_i \in D$ – separated assembly operation, *m* – number of necessary assembly operations.

Every separated assembly operation is implemented by the pre-defined instrument. Also it's described by starting and ending states of assembly details. This expression may be presented in next way:

$$d_i=<I, ndet1, ndet2, kdet1, kdet2>, \qquad (9)$$

where I – instrument,

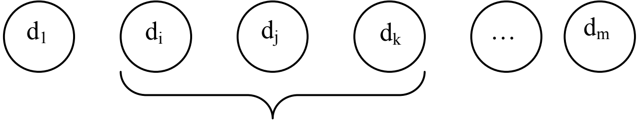
ndet1 – starting position of detail1,

ndet2 – starting position of detail2,

kdet1 – ending position of detail1,

kdet2 – ending position of detail2.

The standard assembly process subsequence <d_i, d_j, d_k> may be found in the scheme AS(D)$_i$, it will represent the standard technological assembly operation (TO) or, in other words, the technological operation framework. The example of standard technological operations (frameworks) extraction in the technological process structure is presented on the fig.2.



Frame	Put_diodes	
Instrument	Robot1	
Detail1	Diode1	
DetailN	DiodeN	

Fig. 2. Extraction of subsequence in technological process

Notice, that setting of every type of elements on board, requires a whole set of actions – subsequences of technological process. For example, framework "Set_diods" includes operations, necessary to set the diodes and includes: operations of taking of diode from the case, its movement to the necessary place on board, orientation around control points and, at last, setting diode to the defined place. Analogically there are frameworks for any type of attached elements: microchips, condensers, transistors, resistors, inductors etc.

Also during process execution, the standard subsequences of assembly process should be ordered on execution order. I t can be done as to criterions of time and materials economy, so we have to find subsequences, that may be executed parallel, to minimize the orientation of robot's manipulator, details and instruments during assembly process.

VI. The description of voice input information method

The method of voice information input is a description of operations, which can be used for the computer-aided assembly technological processes design in robotized manufacturing. In particular, it assumes that assembly technological processes design for the industrial robot with the voice information input is based on the next principles.

1. The assembly unit's positions in the robot's workspace determine the issue and sequence of the technological and auxiliary movements. In terms of predicate logic it can be written in the following way:

$$\begin{aligned}&\forall x(\mathrm{type}(x,\mathrm{assembly\_detail}))\\&\exists y(\mathrm{type}(y,\mathrm{technological\_step}))\\&\exists r(\mathrm{type}(r,\mathrm{rule}))\rightarrow\\&\exists \vec{t}(\mathrm{type}(\vec{t},\mathrm{technological\_steps\_sequence}))\end{aligned} . \qquad (10)$$

2. The coordinates of any point, any command on robot movement can be expressed by the voice in limited natural language. This expression may be represented in a next way:

$$\begin{aligned}&\forall p(\mathrm{type}(p,\mathrm{point}))\forall c(\mathrm{type}(c,\mathrm{command}))\wedge\\&\wedge\exists vp(\mathrm{type}(vp,\mathrm{voice\_present})\end{aligned} . \qquad (11)$$

3. The commands, sequence of which provides the execution of goals of technological or auxiliary movements, can be united to meta-commands. This expression has next presentation:

$$\begin{aligned}&\exists c(\mathrm{type}(c,\mathrm{command}))c\in y(\mathrm{type}(y,\mathrm{technological\_step}))\\&\wedge y\in T(\mathrm{type}(T,\mathrm{common\_technological\_step}))\rightarrow\\&\exists mc(\mathrm{type}(mc,\mathrm{metacommand}))\end{aligned} \qquad (12)$$

4. The control commands sequences, set by voice, implement the separated technological or auxiliary movements, the separated technological operations and define the whole technological process. This expression is represented by next group of expressions:

$$\begin{aligned}&\exists vp(\mathrm{type}(vp,\mathrm{voice\_present}))\rightarrow\\&\exists\vec{VP}(\mathrm{type}(\vec{VP},\mathrm{voice\_commands\_sequence}))\end{aligned} , \qquad (13)$$

$$\exists\vec{VP}\rightarrow\exists\vec{t}(\mathrm{type}(\vec{t},\mathrm{technological\_steps\_sequence})), \qquad (14)$$

$$\exists \vec{t} \rightarrow \exists \vec{to}(\mathrm{type}(\vec{to}, \mathrm{technological\_operation\_sequence})) \quad (15)$$

$$\exists \vec{to} \rightarrow \exists \vec{tp}(\mathrm{type}(\vec{tp}, \mathrm{technological\_process})) . \quad (16)$$

5. The order of voice input application using is set on base of assembly drawing and defined by the models of assembly units and technological process presentations.

6. The voice information input provides the functioning of input information subsystem for Numerical Control (NP) CAD of robot.

The usage of considered voice input information method requires following the next steps.

1. Set the assembly details in the workspace of robot.
2. Using the voice information input, learn robot for the control point's set, defined by location of assembly details, use commands "rotate", "move", "fix".
3. Join complex commands complex, into meta-commands.
4. Using the voice information input, set the necessary robot's movements, representing technological steps and operations.
5. Check the formed technological process, repeated it. Correct if necessary.

After all, the control system of robot should provide the execution of real commands for joints of robot.

VII. Decision-making for manipulation tasks

As it was written in previous section, it's quit useful to define meta-commands. Our understanding of meta-commands is following: the technological process (as to definition) consists of technological operations, in its turn every operation consists of technological movements. Technological movements in case of assembly operation can be like that:

- take Object at point X,Y,Z;
- take Object A (at predefined position);
- put Object A on Object B;
- put Object A to point X,Y,Z;
- change Object A and Object B;

and more simpler actions:

- close Gripper;
- open Gripper;
- move to point X,Y,Z;
- rotate <base> - 90.

It clear, that the listed movements can be set or by keyboard, or by voice information input method, proposed above. From other hand, not every movement (action) can be implemented without any link to others. For instance, action put Object A on Object B assumes, that manipulator arm:

- firstly checks (maybe by technical vision device) if there is free access to Objects A and B, if no access – special procedure is required;
- moves to location of object A (action consists of several movements rotate of move);
- open Gripper (then time pause);
- time pause;
- close Gripper (then time pause);
- moves to location of object B (action also consists of several movements rotate of move);
- open Gripper;
- time pause;
- move to initial location or next program procedure.

The subject of question – how robot will "understand" the essence of such sequence. Certainly this task can be decided by methods of artificial intelligence, or namely – by application of decision-making (planning) systems [4].

The implementation of decision-making systems assumes the presence of planning and executive subsystems (manipulator or mobile robot). The environment of such systems contains the model of workspace, the set of operational schemes and the goal (or goals).

Actually, every operator scheme is a description of action, which can be executed by robot with objects within workspace. So, the mentioned technological movements, from decision-making point of view, should be presented as operational schemes set.

Every operational scheme contains the next elements:

1. the expected result of operational scheme (according to which, the planning system makes it's action selection);
2. the test of achieved goal (possibly, action is already done);
3. validation test (is selected scheme suitable for goal);
4. the list of required actions (the setting and execution of actions to support the considered operational scheme);
5. the list of deleted events (obsolete for scheme result);
6. the list of added events (added by scheme result).

The goal of decision-making system is formulated as expected fact, which should be achieved by system (and with is a parameter for special components – "decider"), for example in case if robot arm should take object can be set as follows: plan (arm_robot (object1, now)).

From the knowledge base point of view, such operational scheme corresponds to framework model and is a framework-scenario. Another key feature of framework-scenario is a definition of order for frame execution slots. Also, according to this model, the attached functions (daemons) have to consider.

The example of operational scheme for "Take Object" action is shown in Fig. 3.

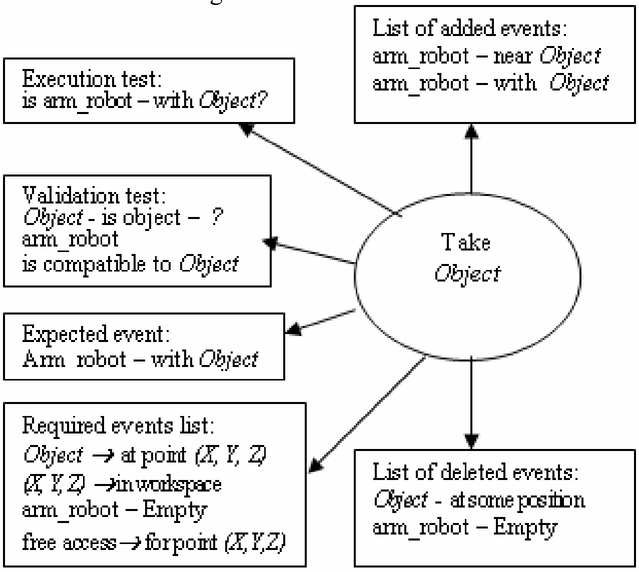

List of added events:
arm_robot – near *Object*
arm_robot – with *Object*
Execution test:
is arm_robot – with *Object?*
Validation test:
Object - is object – ?
arm_robot
is compatible to *Object*
Take
Object
Expected event:
Arm robot – with *Object*
Required events list:
Object → at point *(X, Y, Z)*
(X, Y, Z) →in workspace
arm_robot – Empty
free access→for point *(X,Y,Z)*
List of deleted events:
Object - at some position
arm_robot – Empty

Fig. 3. Operational scheme for decision-making system

Other level decision-making system is adaptation, including compute vision methods. Here is a number of problems with initial information processing, it's recognition and identification. However, the key moments are also knowledge representation, determined by object features:

- the classification of object;
- object state;
- absolute or relational object coordinates;
- object priority level.

Therefore, every object can be classified to some framework description, included to database of decision-making system.

VIII. Conclusion

Probably, the look to the nearest development of assembly technological processes description will include the 3-dimensional description of a projected product, which provides more evident and efficient simulation of objects. Simultaneously, it'll be impossible to exclude the using of assembly processes logical laws for basic technological features extraction. The expression of such technological features using presented logical model is the most simple and preferable way. Using of logical models in CAD and also in the intellectual assembly robot decision-making support system looks especially natural. The analyses of connections on assembly construction can be implemented by semantic, the standard suborders – by framework models.

As the basic proposals of given article there should be noticed:

- logical model for assembly technology CAD representation;
- semantic model for assembly elements analyses;
- framework model application for assembly technology description is defined;
- method of voice information input assembly CAD of robot is proposed;
- framework model applications for operational schemes description in robot manipulation decision-making system are described.

The application of developed models and method for assembly robotic technology has affordability 1000 GBP / year for one robot work station.

References

[1] Chelitshev B.E., Bobrova I.V., Gonsales-Sabater A. Design Automation of Technology in Machine-building. – Moscow: Machine-building, 1987. – 264 p. (in Russian).

[2] Yampols'kiy L.S., Lavrov O.A. Brain-like intellect in industry planning and management. Kiev: High Scool, 1995. – 254 p. (in Ukrainian).

[3] Van King Tan, Eugeniusz Eberbach: A CSA-based Robot Plan Generator // ACM Conference on Computer Science, Indianapolis, IN, USA, 1993. – P. 115-123.

[4] George F. Luger. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Pearson Education. 2008. – 784 p.

[5] John Malpas. Prolog: a relational language and it's applications. Prentice Hall, 1987. – 464 p.

Igor Sh. Nevlyudov. Date of Birth: 25.11.1940. Education: Kharkiv Mountain Institute, Kharkiv (1967), Candidate of Technical Sciences (PhD) (1975), Doctor of Technical Sciences (1986), Professor, Technology and Automation Department, Kharkiv Institute of Radio Electronics (1986), The chief of Technology and Automation Department, Kharkiv National University of Radio Electronics (1994).
His main publications: Hardware and Software for Digital devices Diagnoses on Technological equipment (co-author, 2008), 216 p.; Computer technologies of automated Industry (co-author, 2007), 368 p.
Prof. Nevlyudov is a member of Applied Radio Electronics Academy.

Olexander M. Tsymbal Date of Birth: 11.10.1970. Education: Kharkiv State University, Kharkiv (1992), Candidate of Technical Sciences (PhD) (1996), Assoc. Professor, Technology and Automation Department, Kharkiv National University of Radio Electronics (2002).
His main publications: Visual C++: Programming Technologies (2006), 336 p.; Decision-Making Systems for Intellectual Robots (2004), Eastern European Journal of Modern Technologies № 4 (10), pp 60-63.

Svetlana S. Milyutina. Kharkiv, Date of Birth: 16.12.1982. Education: Kharkiv National University of Radio Electronics, Kharkiv (2005), Assistant, Technology and Automation Department, Kharkiv National University of Radio Electronics (2008).
Her main publications: Neural Nets Application for Voice Information Input in Technological CAD of Robot (2007), Radio Electronics and Informatics № 1, pp. 56-61, Robot Control Commands Voice Form in Technological Process CAD (2008) Eastern European Journal of Modern Technologies № 2/2 (32), pp 60-63.