SYNTHESIS OF SOFTWARE APPLIANCE MACHINERY OF MAP CARD IN MACHINE-BUILDING MANUFACTURE

Sobchak Andrii[1], Kvasnikov Volodumir[2]

[1]National Aerospace University, "Kharkiv Aviation Institute", Ckalova str., 17, Kharkiv, 61070, Ukraine

[2]National Aerospace University, Prospect Kosmonavta Komarova, 1, Kiev, 03680, Ukraine

Abstract

The article proposed the automation technology of the stage of the product life cycle of "technological preparation of the production of products" of a machine-building enterprise. An approach to the conversion of technological documentation in the form of a route map for manufacturing parts of the product, the structure of the software and hardware electronic compiler, and the language that this device implements in hardware is selected and proposed. The proposed technology contributes to the creation of smart enterprise, which is one of the necessary conditions for the competitiveness of engineering production with the advent of the era "Industry 4.0"

Keywords: *Route flow charts, software and hardware compilation, text conversion, smart-enterprise*

1. INTRODUCTION

Global automation of manufacturing processes in machine-building manufacture led to the emergence of machine tool with numerical control, which was traditionally implemented on the basis of personal computers, adopted in the last century to be called electronic computers.

Information on the management of the robotic complex, which is a machine with numerical control to perform the operation of the technological process of manufacturing parts of the product engineering production was entered using perforated cards and then on magnetic carriers. This was a necessary and sufficient condition at the development stage of the production of «INDUSTRY 3.0.» However, with the advent of the development stage of the «Industry 4.0» production, numerical program control in the traditional form for robotic industrial complexes remained a necessary but insufficient condition for constructing smart enterprises in geographically spatially distributed production elements in the form of machines, equipment and complexes.

2. MATERIALS AND METHODS

2.1. Materials

The materials are the transformation of texts, signs, which are physical entities, intended mainly for adaptation of technological documentation (route maps) with technical means. The attitude of adaptation of technological documentation (A and B) with technical means simplified expressed by the following scheme D. V. Soshnikov (2003, p. 19):

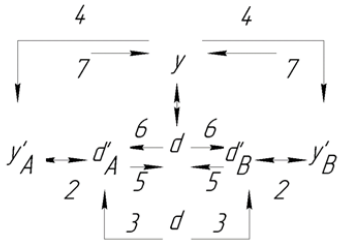


Fig. 1. Scheme of communication attitude

Source: D. V. Soshnikov, 2003, Perspective Information Technologies and Intelligent Systems, Vol. 3, no. 15, p.19.

The following symbols are depicted in the figure:

y – object marked with a sign;

d', y' – images of the sign and the object in the route map.

The arches in the scheme fix different relations. Thus, relations 1, 2 describe the semantics of the sign d and its image d '; 3, 4 - the appearance of images of the sign d and the object in; 5, 6 - formation of signs and their reflection; 7 - the impact of technological documentation on production.

For control systems, a set of functions performed by a sign system can be represented as a contour (Fig. 2).

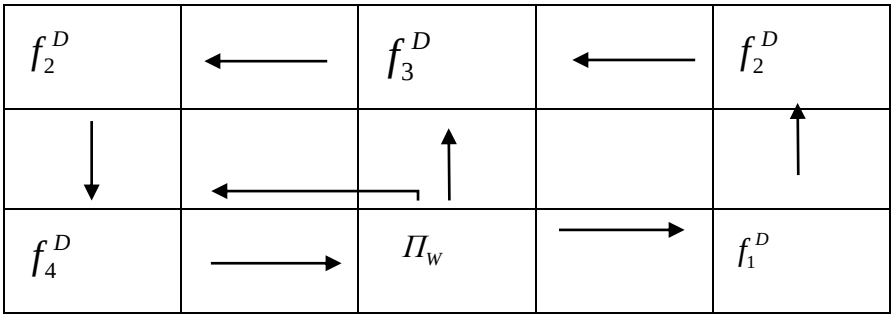


Fig. 2. A set of functions of control systems in the form contour

Source: D. V. Soshnikov, 2003, Perspective Information Technologies and Intelligent Systems, Vol. 3, no. 15, p.21

Figure Π_w – Object of Management (OM), which operates based on the digital control program firmware (DCPF) machine with numerical control (MNC); f_1^D – formation of route maps describing the required state of the (OM); f_2^D – conversion of route maps, which does not require the display of images of objects described by them; f_3^D – transformation of route maps into the state of the OM and the formation of the deregister for the decisions made on the OM; f_4^D – - mapping of maps in the actions of the OM taking into account the influence of the DCPF machine MNC.

Functions can be performed by a person, either by technical means, or jointly - in a human-machine mode. Accordingly, there are varieties of characters perceived by human or technical means, as well as signs that are perceived by both man and machine.

2.2. Methods

The methods are asked by means of instructions, that is, route maps describing the sequence of actions for the implementation of certain functions. For technical means, the role of the method is performed by a machine program, in the form of a DCPF, in accordance with a routing technological map (text) containing a sequence of actions that must be reproduced in a machine at the subcontractor's production. In addition, the method can be implemented as a special device.

2.2.1. Ways for converting the information

The main function of sign systems is the conversion of texts. Consider the subclass of single-word transformations of texts, which change: the type of characters, the carrier of signs, the place of the sign on the carrier, the space carrier in space, storage time. Possible composition of these transformations can be established using Table 1, which distinguishes the following types of texts, respectively, the varieties of their carriers and signs: language D^p graphic symbols D^C perforated D^n and ecorded on magnetic carriers D^M .

Input D_1	Output D_2		
	Signs Symbols	Symbols	Symbols on magnetic carriers
Language Signs	Language Signs Conversation	Writing Speech Text	Record of broadcast on a magnetic film
Symbols	Reading aloud	Rewriting, encoding, reproduction	Record on the magnetic carriers from the document
Symbols on magnetic carriers	play recording	Printing text from Magnetic Carriers	Rewriting

Table 1. Conversion in a sign system

Source: D. V. Soshnikov, 2003, Perspective Information Technologies and Intelligent Systems, Vol. 3, no. 15, p.22

In each cell of the table (except for the diagonal), the transformations associated with the replacement of the type of carriers and characters indicated on the line are presented in the form, which is determined by the column of the table. The cells on the diagonal correspond to transformations, in which the copies of the carrier, signs, their places on the carrier change. Examples of them are: rewriting the text, displaying the text on the screen, reproducing (changing the media), sorting (changing the place of mark on the media), encoding (replacing one character to another). One of the varieties of transformations is the change of the place of the carrier in space, for example, the transmission of texts in the system from one workplace to another. A dash in a cage may indicate a lack of technical feasibility or appropriate conversion. The need for the storage of texts is caused by the discrepancy between the moments of time of their receipt from sources and use. Storage can be considered as a text conversion, which changes only the duration of its stay on a particular medium. This transformation is served by two procedures - making changes and extracting (copying) for use.

In design and control systems, the processes of converting texts are lined up in technological networks or, in a special case, in a chain of transformations, for example:

$$D_{1p_1}^{\text{С1}} f_2^{CC} \rightarrow D_{1p_2}^{\text{С2}} f_2^{CП} \rightarrow D_{1p_3}^{\text{П}} f_2^{ПМ} \rightarrow D_{1p_4}^{\text{М1}} f_2^{MM} \rightarrow D_{1p_5}^{\text{М2}} f_2^{d} \rightarrow D_{1p_6}^{\text{М1}} f_2^{MM} \rightarrow D_{2p_7}^{\text{М2}} f_2^{MC} \rightarrow D_2^{C}, \quad (1)$$

where D_j^{ik} - the i-th type, represented on the k-th media at the input (j = 1) or output (j = 2) of the sign system;

P_m - sign characteristic of the converter;

$f_2^{i_1 i_2}$ the function of the converter of characters from the i1-th type in the i2-th type, is determined by the table 1 (i = c - symbols, i = п - punctuation sings, i = m - sings on the magnetic carriers).

When you add in technological processes control circuit formed backflow signs for error correction.

The relationship between the characters in the text is called syntactic. The simplest of these is the relationship of characters in linear texts, in which for each sign d_i there is a previous sign (with the exception of the first one) and further - d_{i+1} (with the exception of the latter). In non-linear texts, each sign can be associated with several characters - for example, in the chemical formula, tree-like structure, network, mathematical formula, document. For machining, nonlinear text can be converted to linear. But such a change worsens the perception of the text by man and is associated with an increase in the volume of the carrier. Since the characters are represented on the carrier, relations between them can be set not directly, but with the help of a certain placement on the carrier. Such a relationship is called topological. Therefore, for the formation of texts developed rules for placement of characters on the carrier, and for their perception - the rules of reading.

When converting texts, it is necessary to search for them the necessary characters, words, phrases, parts of the text for copying, replacing or deleting them. Different search methods are related to the topological properties of texts, such as the order of placement of characters. One of the procedures that require the search for characters is the text encoding D_1 , for which for each sign $d_1 \in D_1$ need to find it replaces the sign $d_1^* E D^*$, resulting in a D_2 text with new characters. The text D^* is a bilingual alphabet $D^* = \{(d_1^*, d_2^*)\}$, where it serves as the search key, that is the d_1 sign is compared with the characters d_1^* , until it is found $d_{ij}^* = d_1$. Then sign d_1 is replaced by $d_2^* \in (d_{ij}^*, d_2^*) \in D*$

The coding procedure can be organized in the opposite way - a sign d_1^* in the text D_1 corresponds to the identical sign d1, plays the role of the key. With this option, it is possible to arrange the text D_2 in the same way as D^*, but the cost of searching time here will be much larger than in the first variant (if the ordering of texts is different). The procedure for changing the ordering of the text is called sorting. It is designed to accelerate the search: in the unregulated text length N the length of the search for a character $L \approx 0.5N,$ in order - $L \approx \log_2 N$.

The cost of this acceleration is the cost of sorting time. They are justified for multiple subsequent searches in orderly text. The processes of converting texts also includes their control, consisting in a symbolic comparison of two texts to identify inconsistent characters with the same value of the key, in comparison of numerical values with restrictions, in checking a given length of sign collections, which requires the text in their separators, and so on.

3. RESULTS

Sobchak A. P. argues (2004, 2005), that procedure for encoding the converted information from the routing technological map: the use of special microprograms for building systems greatly increases the resource and provides the necessary flexibility of software tools. The speed of hardware in many cases is much higher than necessary for practice. This can reduce clock speeds and increase the reliability of the hardware compared to the software implementation.

Thus, the required level of flexibility of the automated system and its reliability is a function of hardware support for recursive automata with programmable logic (RAPL). Based on the approach Zhikharev (1998) and Lelyuk (1990) experimental testing of the required level of flexibility is one of the tasks of the RAPL layout. Input of the research was developed model of the electronic compiler, which is presented in Fig. 3.

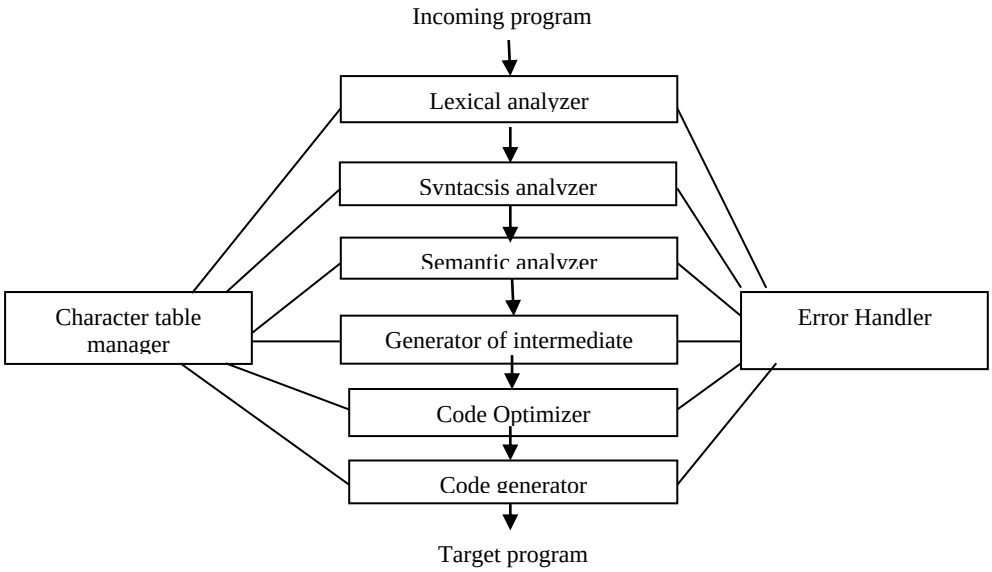


Fig. 3. The structure of the electronic compiler

The functional structure of the electronic compiler (EC), developed as part of the RAPL layout, is shown in Fig. 4, where the following structural levels are allocated in the functional plan:

- software, which includes the standard control of the control panel and means of implementing unique functions;

- hardware, consisting of EC and communication devices with the base PC;

- service, software and hardware for user input languages, configuration of operating and control automata, electronic compiler and user interface.

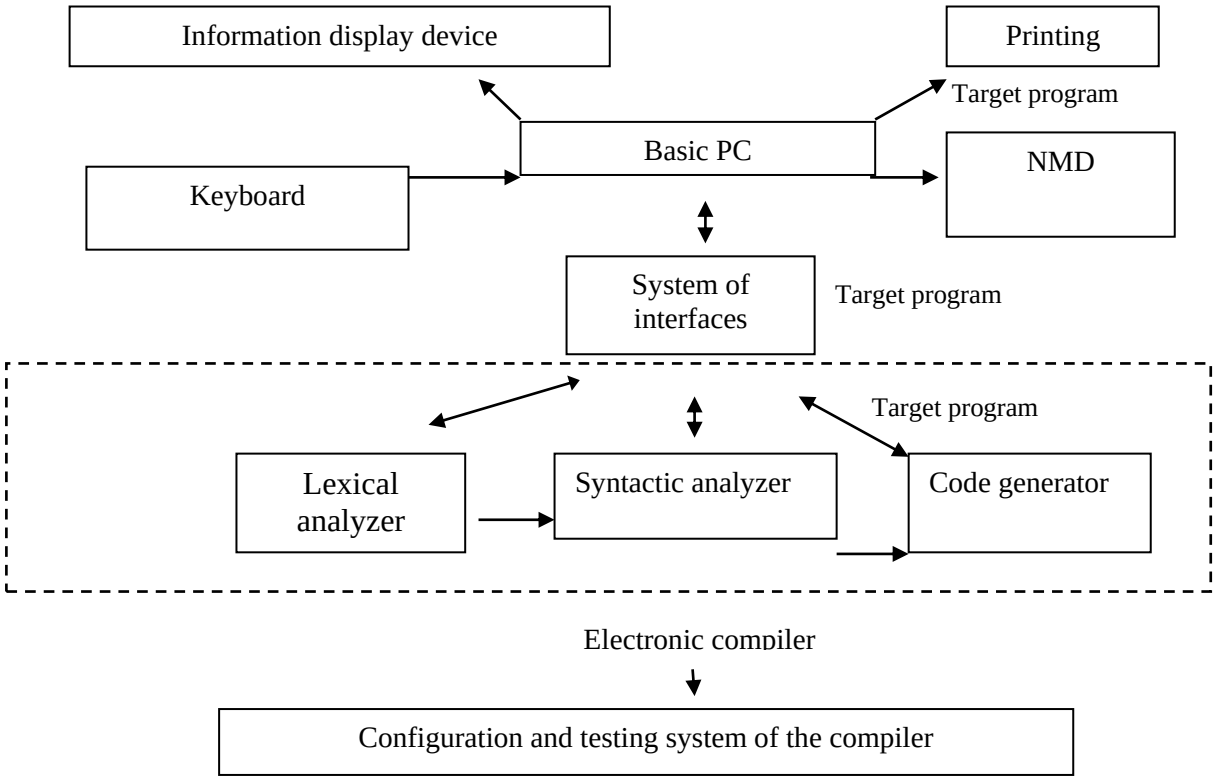


Fig. 4. The functional structure of the electronic compiler

Aho (2001) uses the following term in his works "electronic compiler" - the core of RAPL, a system of three specialized processors: a lexical analyzer, a parser, and a code generator. These processors are executed in the form of autonomous blocks, functionally and informatively interconnected.

Lexical analyzer (LA) allocates in the input text of the program separate tokens in the form of 16-bit codes. Syntactic analyzer (SA) generates syntactic control of the program and detects possible errors.

The electronic compiler is constructed according to the schema of syntactically controlled translation according to the grammar of the language given using syntactic diagrams and BNF. In this regard, the electronic compiler adapts to the constantly changing languages and its diagnostics considerably simplifies. The rest of RAPL devices are synthesized in accordance with the principles of constructing an electronic compiler based on systems of interpreted algorithmic algebras using the synthesis of automata that are part of the RAPL structure and represent high-level languages.

One way of accelerating information processing and improving the efficiency during simulation is to create automated systems Sobchak believes (2013) that meet the requirements of configuring databases and knowledge bases to solve a problem. For the design of automated systems, recursive systems of interpreted algorithmic algebras (R-SIAA) are proposed, which are also used to develop a technology for the synthesis of recursive automata with programmable logic (RAPL).

RAPL provides compilation of the text of the functional specification in the simulation algorithm or in the temporary diagram, which reflects the change of the logic signals of the electronic circuit of the functional block (or the change of microcomputers entering the general bus of the built-in object of the controller of the microprocessor system).

One of the most important problems is the problem of compiling programming languages and formal languages in general. The process of compiling languages consists in the implementation of the functions of lexical and parsing analysis, generation of executable code. For the formation of formal

languages presented in the Bekusovsky normal form (BNF) or syntactic diagrams (SD), the theory of formal grammars has become widespread. It is known that languages largely determine the level of technology design systems. For example, Pascal corresponds to structural design, C ++ - object-oriented design, etc. In the creation of machines with structural interpretation of languages, unlike software compilation, they perform circuit (hardware) methods based on electronic compilers, the speed of which is 2-3 orders of magnitude higher than PC.

Elemental base RAPL is the usual chips that are manufactured by industry. The set of these chips, organized in a special special way, is RAPL, the abstract model of which is the composition of the control and operating structures (Fig. 5), where the following designations are adopted:

A - management structure (CS);

B - operational structure (OS).

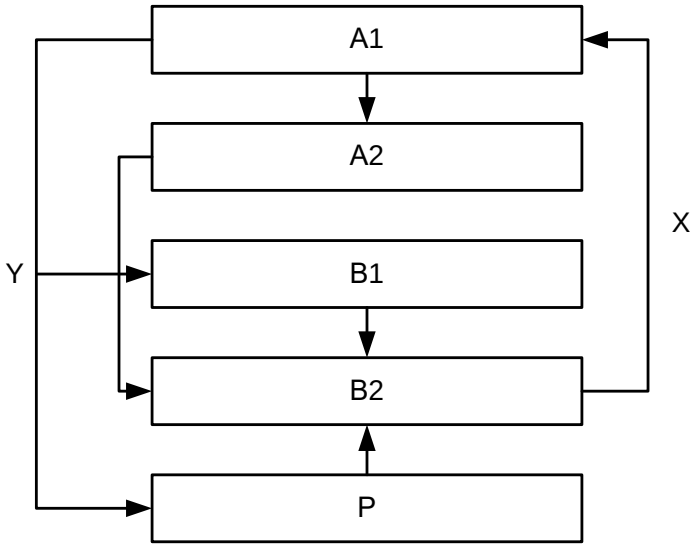


Fig. 5. RAPL structure

Let $G_0 = \langle N, \Sigma, S, T, H \rangle$ – a formal grammar that generates an arbitrary formal language *L (G₀)*,

where Σ - is the terminal alphabet;

N - is a non-terminal alphabet of grammar *G₀* which has an axiom *T* and a circuit of output *N*;

S - set of synchronous terminals;

A recursive automaton with programmable logic is called the structure:

$$A = \{A_1, B_1, X, Y, \delta, a_1, S, G_0, F^*(A_1), Z\}, \quad (2)$$

where $A_1 = \{a_i\}, i = \overline{1, g}$ – - the set of states CS;

$a_1 \in A_1$ – initial state of CS;

B_1 – set of OS states;

$X = \{x_m\}, m = \overline{1, M}$ – set of input lexical signals CS;

Y – set of initial states of RAAP;

δ – some function of displaying sets depending on A_1, Σ, N;

$F^*(A_1)$ – a subset of the final states of the CS;

Z - is an information component of the output function OS.

The volume of RAPs potentially depends on the complexity of the grammar presented in a corresponding algebraic form.

4. DISCUSSION

Until now, the traditional classical approach was a software method of compiling technological documentation. In this case, standard computers interfaces (RS-232, RS-454, etc.) were used, which introduced errors in data processing, as well as a loss in time, since it was necessary to adapt standard, often unused computer nodes to direct control processes or processes with feedback. Also, the complexity of software compilers was determined by the need / presence of a user interface, which entails the presence of additional compilation from higher-level languages to low-level languages understood by the processor of a personal computer. All these circumstances led to the complication of creation, especially for remote control of technical processes, of a flexible smart-structure of the production process.

5. CONCLUSIONS

Thus, unlike the known discrete model of the discrete converter (DC), the KDP operates not only according to the rules of finite automata, but also according to some grammar GO, which can be automatic, context-free, context-dependent or grammar of the general form.

This difference is significant and predetermines the relevance of the described technology of software and hardware compilation in the construction of smart-enterprises and is a new constructive boost to the synthesis of compilers.

REFERENCES

1. Soshnikov DV 2003, "The Architecture of a Distributed Framed Hierarchy for the Construction of Intelligent Systems for Distributed Accumulation and Reuse of Knowledge", *Perspective Information Technologies and Intelligent Systems*. Vol. 3, no. 15., pp. 18-22.
2. Sobchak AP, Ilyushko VM, Khodarev KV, Smirnov N. D. 2005, "Problems of creating a hardware compiler of elementary mathematical functions", Aviation-space technology and technology, Vol. 7, no. 23, pp. 294-303.
3. Sobchak AP, Khodarev KV 2004, "*Implementation of the model of the parser in the language of the hardware description*", Open Information and Computer Integrated Technologies, Vol 25. pp. 90-93.
4. Zhikharev VY 1998 "A generalized model of the symbol process", *Open informational and computer integrated technologies*, Kharkiv. aviation them N. E. Zhukovsky. Kharkiv, Vol 1., P. 138-144.
5. Lelyuk VA 1990, *Conceptual design of systems with knowledge bases*, Basis, Kharkiv.
6. Aho A, Networks R, Ulman D. 2001, *Compilers: Principles, Technologies*, Williams, Moscow.

7. Sobchak AP 2013, "Application of Artificial Intelligence in Mechatronic Systems", *Electrical Engineering and Electromechanics*. no. 2. pp. 24-31.
8. Sobchak AP, Smirnov ND 2005, "*Hardware implementation of control device compression device on programmable logic integrated circuits"*, Radioelectronic and computer systems. Vol. 3, no.11, pp. 34-39.