SYNTHESIS OF STRUCTURAL ELECTRICAL CIRCUITS OF RADIO ENGINEERING DEVICES IN A HYBRID PRODUCTION EXPERT SYSTEM

Georgy A. Dolin, Anastasiya Y. Kudryashova,
Moscow Technical University of Communications and Informatics, Moscow, Russia,
dolin1974@gmail.com, asykka@bk.ru

DOI: 10.36724/2664-066X-2020-6-2-5-9

ABSTRACT

The development of communication systems and devices requires full automation of their design process to quickly update the RED. Especially important is the development of software for the synthesis of basic electrical circuits. The article describes the algorithm of end-to-end CAD based on an expert system for the synthesis of basic electrical circuits of RED based on an object-oriented hybrid expert system. Algorithms for forming the knowledge base at the learning stage and output in the synthesis process are considered. The algorithmic and software hybrid production ES and knowledge base are described. The system organization of structured information about the synthesis of blocks and the entire RED. It allows you to effectively form and manipulate the knowledge of RED design experts. The method for introducing and using a set of constant and variable confidence coefficients in the ES has been developed, which allows using unformalized knowledge about the field of RTU design. This ensures the application of both the knowledge obtained during the training of the ES and the knowledge accumulated during the design of the RED of the ES itself. All this allows the designer to formalize knowledge faster and more accurately, as well as increase the speed of automatic design since unlikely circuit solutions are not considered.

KEYWORDS: *expert system, Radio Technical Devices, CAD, knowledge base, algorithmic structures*

INTRODUCTION

Analysis of the accumulated experience in the field of Radio Engineering Devices (RED) design automation has shown that it is possible to ensure the synthesis and modeling of RED of any structure with a high degree of reliability and optimality only by creating a CAD system containing an ES that allows you to present and programmatically implement empirical knowledge, as well as heuristic rules and techniques used by highly qualified specialists in the design of RED in traditional manual design [1].

Expert synthesis allows you to provide a lack of a priori information with greater unambiguity and reliability of decisions made during automatic structural synthesis of RED in CAD and the accumulation of design experience of qualified RED design experts. It allows unskilled or novice users to design RED at the level of expert designers by formalizing the collective knowledge of a group of highly qualified experts about the field of RED design.

ES are classified according to the following set of characteristics: purpose; stage of existence; type of PS; type of knowledge representation methods used; universality; basic properties; operating environment. For automated RED synthesis in CAD, it is advisable to develop an ES of the following type: by purpose – research; by type-developed in algorithmic programming languages; by the type of knowledge representation methods used - hybrid, combining symbolic representation of knowledge with mathematical calculations necessary for the design of RED and allowing the use of data stored in external databases in ES; by the type of universality of knowledge representation - with an integral representation, combining several models of knowledge representation; operating in Windows operating systems on a PC.

I. The advantages of ES

ES are designed for solving so-called weakly formalized or unformalized tasks. According to [2], unformalized tasks include those that have one or more of the following characteristics: tasks cannot be set in numerical form; goals cannot be expressed in terms of a precisely defined target function; there is no single target function in the design of RTU, and often for individual nodes; there is no algorithmic solution to problems; the customer does not have complete source information and reliable units of measurement for some parameters; an algorithmic solution exists, but it cannot be used due to limited resources (time, memory, etc.); the amount of source information and data is very large, variable, and requires a significant amount of time for familiarization, which leads to the need to reduce the search space to reduce the design time. ES are used to solve difficult practical problems, which include circuit synthesis of RTU. In terms of quality and efficiency of solutions, such ES are not inferior to the solutions of a human expert, and often surpass them [5] due to the accumulation of knowledge of a group of expert designers and reducing the subjectivity of their decisions. The output process in the ES has clarity, i.e. solutions can be explained to the user at a qualitative level (in contrast to solutions obtained using numerical algorithms and from solutions obtained by statistical methods).

This quality of ES is provided by their ability to reason about their knowledge and conclusions. They can add to their knowledge both in the course of interaction with an expert, and with a novice user in the constructive or non-constructive solution of the problem.

Let's note further the advantages of ES over a human expert:

- ES has no preconceptions;
- ES don't jump to conclusions;
- ES work in a systematic way, considering all the details, often choosing the best alternative from all possible ones.

KB ES can be exceptionally large. Once entered the machine, knowledge is stored forever. A person has a limited KB, and if the data is not used for a long time, they are forgotten and lost forever.

Like other types of computer programs, ES cannot replace a person in solving problems, but rather represent a special kind of tools that allow them to solve problems faster and more efficiently. In addition, knowledge-based systems can be viewed by users as a form of replication - a new way to record and disseminate knowledge.

Therefore, it is possible to provide automatic circuit synthesis and subsequent modeling of RED of any structure only by developing a CAD system containing ES, which will allow forming and storing empirical knowledge used by specialists in the design of RED in traditional manual or automated design, as well as synthesizing constructive RED due to a priori exclusion by design experts of unrealizable and little-promising RED nodes from the KB ES [1].

Selected due to several advantages, the hybrid production ES includes the following main blocks: the KB, the output machine, and the user interface. KB ES consists of facts and rules.

The KB facts describe what is known about the RED synthesis technique now. In the KB of a production ES, facts are represented as a triple: object-attribute-value. The object in RED synthesis is the technical parameters or characteristics of the device that is assigned a value, and the attribute – determines the confidence coefficient for the object value. In KB, rules form a certain hierarchy, which can be represented as a graph consisting of vertices and edges. Each vertex corresponds to an object that has certain values. Each edge is defined by a pair of vertices and corresponds to one of the possible ways to reach the goal. A graph can be represented as a tree whose branches correspond to the edges of the graph, and whose nodes correspond to the vertices of the graph.

There are two groups of requirements for knowledge representation methods for designing RED [2, 5]. the Requirements of the first group of knowledge representation methods assume the following: universality, integrity, and openness of knowledge representation. This group of requirements contributes to improving the efficiency and obtaining high performance characteristics of the developed RED [4]. The second group of methods regulates the functionality of the RED and is crucial for the practical use of CAD. The requirements of the second group imply ensuring the following factors: the adequacy of the display of the subject area, i.e. such a description that can simulate any processes occurring in this subject area and essential for the selected class of problems; the natural form of the description of the subject area in the knowledge system, which allows you to create a human-friendly interface with the computer system in the process of setting and solving design problems [6]; multilevel or hierarchical description of the subject area, which provides solutions to complex design problems; a combination of procedural and declarative methods in a single knowledge system that allows, on the one hand, to simply describe the basic concepts and terminology of the subject area, and on the other - to set functional dependencies and formal [7].

For the synthesis of RED structural schemes, it is advisable to use the production model of knowledge representation about RED, since in the production ES, the conclusion is made in the opposite direction from the statements that must be proved, which significantly speeds up the result; modularity of the rules organization; independence of rules that Express separate unrelated fragments of knowledge; separation of control knowledge from subject knowledge, which allows you to apply various control strategies, including using statistics, for example, the Bayes full probability formula; the ability to create control mechanisms for automatic design of RED. At the same time, the hybrid production ES includes the following main blocks: the knowledge base (KB), the output machine, and the user interface. KB ES consists of facts, questions, and rules.

The KB facts describe what is known about the method of RED synthesis at this moment. In the KB of a production ES, facts are represented as a triple: object-attribute-value. The object in RED synthesis is the technical parameters or characteristics of the device that is assigned a value, and the attribute – determines the confidence coefficient for the object value. In KB, rules form a certain hierarchy, which can be represented as a graph consisting of vertices and edges. Each vertex corresponds to an object that has certain values. Each edge is defined by a pair of vertices and corresponds to one of the possible ways to reach the goal. A graph can be represented as a tree whose branches correspond to the edges of the graph, and whose nodes correspond to the vertices of the graph [2].

The rules in the KB of production ES using the language of representation of algorithmic structures are formulated in the following form:

< rule > :: = IF < premise > then < conclusion >;

< premise > :: = < triple > | < triple > And < premise >;

< triple > :: = < object > < logical operator> < value >, < attribute >;

< conclusion> :: = < three > | <action > | < three > And < conclusion >.

The rule is written as a conditional statement that includes a premise and conclusion. The premise may consist of one or more triples connected by the unions "And". the Conclusion may consist of one or more triples involving one or more actions. An action consists of one of four procedures: calculating a mathematical function, searching the database for an electronic component, saving an intermediate design result, or clearing the object value. An es output machine (rule interpreter) is an ES program block that implements the output process based on a DB and a working set (a part of memory allocated for storing rules and fact values during the output process).

The output machine performs two functions: first, viewing existing facts and rules from the database and adding new facts, and second, determining the order in which the rules are viewed and applied. The output machine performs the design process using rules, stores information about the conclusions obtained for the user, and requests information from him when there is not enough knowledge in the database to perform the next rule.

The RED design process in ES is a sequence of steps, each of which selects a rule from the KB that is used to determine the goal value. The process ends when the fact value is determined or it is shown that it cannot be determined. RED design can be performed in several ways [1], of which the most common are direct output order and reverse output order (depth-first search).

The direct order of inference is made from the facts that are in the working set to the conclusion. If such a conclusion can be found, it is entered in the working set.

In the full iteration method, vertexes are expanded in the order in which they are constructed. A simple tree iteration algorithm consists of the following sequence of steps:

1) Put the vertex in a list called OPEN.

2) If the list is OPEN and empty, the output is signaled that the search failed, otherwise proceed to the next step.

3) ke the first vertex from the OPEN list and move it to the CLOSED list; let's call this vertex n.

4) xpand the vertex n, forming all the vertices immediately following n. If there are no immediately following vertexes, then proceed immediately to step (2). Place the existing vertexes immediately following n at the end of the list of points AND build pointers leading from them back to vertex n.

5) If any of these immediately following n vertices are target vertices, then output the solution obtained by looking along the pointers; otherwise, proceed to step (2).

This algorithm assumes that the initial vertex does not meet the goal, although it is not difficult to enter a step to check this possibility. The vertices and pointers constructed during the iteration process form a subtree of the entire implicitly defined tree of the state space [8]. We will call such a subtree a brute force tree.

The full search method will certainly find the shortest path to the target vertex, provided that such a path exists at all. (If there is no such path, the specified method will declare failure in the case of finite graphs, and in the case of infinite graphs, the algorithm will never finish its work).

There may be problems in which the solution has some other requirements than the requirement to obtain the shortest sequence of operators. Assigning certain prices to tree arcs (followed by finding a decision path that has a minimum cost) meets many of these promised criteria. A more General version of the full search method, called the equal price method, allows you to find in all cases some path from the initial vertex to the target, the cost of which is minimal. While the algorithm described above propagates lines of equal path length from the starting vertex, the more General algorithm described below propagates lines of equal path cost. It is assumed that we are given a cost function c(ni, nj), which gives the cost of moving from a vertex ni to some next vertex nj. In the equal price method, for each vertex n in the iteration tree, we need to remember the cost of the path built from the initial vertex s to the vertex n. Let g(n) be the cost from vertex s to vertex n in the iteration tree. In the case of iteration trees, we can be sure that g(n) is also the cost of the path that has the minimum cost (since this path is the only one).

In the equal price method, vertices are revealed in ascending order of cost g(n). This method is characterized by the following sequence of steps:

1) Put the initial vertex s in a list called OPEN. Put g(s)=0.

2) If the list is OPEN and empty, the output is signaled that the search failed, otherwise proceed to the next step.

3) Take from the OPEN list the vertex for which the value of g has the smallest value, and put it in the CLOSED list. Give this vertex the name n. (If the values

match, choose the vertex with the minimum g at random, but always giving preference to the target vertex.)

4) f n is the target vertex, then output the decision path obtained by looking back in accordance with the pointers; otherwise, proceed to the next step.

5) xpand vertex n by constructing all the vertices immediately following it. If there are none, proceed to step (2). for each of these immediately next (child) vertices ni, calculate the cost of g(n) by putting g(ni)=g(n)+c(n,ni). Put these vertexes, along with their corresponding newly found values of n, in the list of points AND build pointers going back to T.

6) Go to step (2).

Checking whether a certain vertex is a target is included in this scheme so that minimal cost paths are guaranteed to be found.

An equal-price algorithm can also be used to find paths of minimal length if you simply put the cost of each edge equal to one. If there are several initial vertexes, and the algorithm is simply modified: in step (1), all initial vertexes are placed in the OPEN list. If the States that meet the goal can be described explicitly, then the iteration process can be started in the opposite direction, taking the target vertices as the initial ones and using the appeal of the operator G.

In depth-first search methods, the vertexes that were built last are revealed. Let's define the depth of the tree vertex as follows:

– The root depth of the tree is zero.
– The depth of any subsequent vertex is equal to one plus the depth of the vertex that immediately precedes it.
– Thus, the vertex that has the greatest depth in the search tree is currently the one that should be expanded at this moment.

This approach can lead to a process unfolding along some useless path, so you need to introduce some return procedure. After the process builds a vertex with a depth greater than a certain boundary depth, we reveal the vertices of the greatest depth that does not exceed this boundary, and so on.

The depth-first method is determined by the following sequence of steps:

1) Put the starting vertex in a list called OPEN.

2) If the list is OPEN and empty, the output is signaled that the search failed, otherwise go to step (3).

3) ke the first vertex from the OPEN list and move it to the CLOSED list. Give this vertex the name n.

4) If the depth of the vertex n is equal to the boundary depth, then go to (2), otherwise to (5).

5) xpand vertex n by constructing all the vertices immediately following it. Put them (in any order) at the beginning of the list of OBJECTS and build pointers going from them to n.

6) If one of these vertexes is a target, then output the solution by looking at the corresponding pointers, otherwise go to step (2).

The depth - first search algorithm iterates along one path until the maximum depth is reached, then considers alternative paths of the same or lower depth that differ from it only by the last step, then considers paths marked by the last two steps, and so on.

II. The purpose of the ES

The purpose of the ES is to determine the value of the object specified by the designer, or to make sure that with this level of knowledge IN the ES database, the object value cannot be determined. The value of an object can either be pre-defined in the KB of the ES, or it is extracted by the ES from the dialog with the designer [3].

The RED design process in ES is a sequence of steps, each of which selects a rule from the KB that is used to determine the goal value. The process ends when the fact value is determined, or it is shown that it cannot be determined. RED design can be performed in several ways, the most common of which are direct output order and reverse output order (depth - first search).

The direct order of inference is made from the facts that are in the working set to the conclusion. If such a conclusion can be found, it is entered in the working set. Depth-first search: rules are viewed until they are found in working memory or object values are received from the user that confirm one of the rule conclusions. Initially, the designer sets a goal-to determine the design object.

When searching in depth, the inference machine searches for a rule containing the design goal in the conclusion, and then in the process of working with the ES, the inference machine goes back, moving from conclusions to conditions, and tries to find among them those that confirm this proposal. If it turns out to be correct, then the next object is put forward, detailing the first one, which is a sub-goal in relation to it.

Next, we look for conditions that confirm the truth of the subordinate conclusion. Search in depth is used in cases where the goals are known and relatively few of them, which is most consistent with the method of synthesis of RED structural schemes. In addition, this method of achieving the goal allows you to speed up the design process by introducing confidence coefficients to select the further path of movement along the tree, i.e. reducing the search space in width, and taking into account the variance of confidence coefficients to reduce the search space in depth. Therefore, due to the noted analogies and advantages, this method of achieving the goal is chosen for the implementation of the process of synthesis of structural schemes of RED in ES [5].

To implement the circuit synthesis of RED in the output machine of the production ES, at least the following procedures should be implemented: FIND, SOLVE, SAVE, and CLEAN.

To find the required electronic components MOUTH on specified parameters to a procedure FIND (Query_SQL), you must pass the SQL query of the following form: choose all components from the database, where the values given by an expert or identified in the process output parameters of the required electronic components more or less equal to the values of the component parameters in the database.

The result of this procedure is the selection of electronic components whose parameters meet the constraints specified in the request. If the query results in multiple components, the designer is asked to select one of them, or the ES automatically selects the first appropriate one. To calculate the values of mathematical expressions that contain integers and decimals, parentheses, signs of four arithmetic operations, and mathematical functions such as exponentiation, root extraction, integration, differentiation, and so on, and are written as a string, you need to implement their parsing (the SOLVE/EXPRESSION procedure).

To do this, it is advisable to use the Polish form of writing (Backus notation) mathematical expressions. The SAVE (list of objects) procedure allows you to save the design results in a text file, and the CLEAR (object) procedure allows you to clear the object value defined during the design process.

The algorithm for synthesis of RED structural schemes in a hybrid production ES includes the following steps. The analysis of the TOR includes entering the parameters of the TOR by the designer and forming the design goal. If the designer does not have the values of the required parameters, the ES cannot carry out the design process. And if it is necessary to take into account additional requirements of the TOR, the designer should make changes to the KB ES. At this stage, the reliability of heuristic information stored in the database and necessary for constructive synthesis is of particular importance [4].

The development of the device at the structural level includes the selection of the nomenclature of nodes and cascades of the structural scheme of the designed RED from the KB. The hybrid production expert system allows you to present methods for the synthesis of RED structural schemes in the form of rules. They make it possible to save the conclusions obtained for the user and request additional information from him, calculate the required parameters of structural schemes based on functional relationships, and select electronic components based on several parameters from the database. The synthesis of structural schemes is carried out in a hybrid production ES that operates with symbolic information in combination with formula relations. Structural design ends with the formation of requirements for the parameters of individual RED nodes, which should ensure the operation of the entire device.

Conclusion

Thus, the hybrid production ES allows you to implement expert circuit synthesis of structural diagrams of individual nodes and the entire RED as a whole, due to the fact that the output process is similar to the process of reasoning of an expert designer. In addition, the ES provides the calculation of the required parameters of the RED block diagram by functional relationships and it is possible to select electronic components based on a set of parameters from the database. At the same time, the use of confidence coefficients for choosing the design direction and taking into account the variance of confidence coefficients allows you to speed up the design process by reducing the search space in depth and width.

References

1. G. A. Dolin, "Object-Oriented Representation of Mixed Models Knowledge in the Design of Electronic Devices in CAD Electra," *2020 Systems of Signals Generating and Processing in the Field of on Board Communications*, Moscow, Russia, 2020, pp. 1-5, doi: 10.1109/IEEECONF48371.2020.9078546.

2. G. . Dolin, "Using a distributed database of parameters of electronic components in course design," *Methodological issues of teaching infocommunications in higher school*, 2018. No 2. pp. 10-14.

3. G. A. Dolin, "Formation the knowledge base of red expert design for conducting coursework and lectures," *Methodological issues of teaching infocommunications in higher school*, 2018. No 2. pp. 46-53.

4. G. A. Dolin, "Circuit Analysis, Synthesis and Simulation of Radio Devices in Electra CAD," *2019 Systems of Signals Generating and Processing in the Field of on Board Communications*, Moscow, Russia, 2019, pp. 1-6, doi: 10.1109/SOSG.2019.8706814.

5. G. A. Dolin "Automation of synthesis of structural and basic electrical circuits of RED in production and object-oriented expert systems," *2018 Actual problems of radio and film technologies. II all-Russian scientific and technical conference.* Saint Petersburg, Saint Petersburg state Institute of film and television, October 24-27, 2018. Collection of works. pp. 40-45

6. K. V. Boychenko, J. Rivory, I. V. Boychenko, A.Y. Kudryashova, "Interactive Space as a Signal Processing Device," *2020 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)*, Svetlogorsk, Russia, 2020, pp. 1- 10.1109/SYNCHROINFO49631.2020.9166033.

7. K. V. Boychenko , I. V. Boychenko, A.Y. Kudryashova, "Interactive built environment in shaping users orientation and navigation in space ," *2020 Systems of Signals Generating and Processing in the Field of on Board Communications*, Moscow, Russia, 2020, pp. 1-4, doi: 10.1109/IEEECONF48371.2020. 9078658

8. K. V. Boychenko, I. V. Boychenko, A. Y. Kudryashova, "Interactive Built Space as the New Means of Information Communication," *2019 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)*, Russia, 2019, pp. 1-4, doi: 10.1109/SYNCHROINFO.2019.8813912.