AN EXPERT SYSTEM APPROACH FOR THE DESIGN OF CABLE NETWORKS

Iñigo Monedero
School of Computer
Science and Engineering
Electronic Technology
Department
Avda, Reina Mercedes s/n
41012 Seville (Spain)

Carlos León
School of Computer
Science and Engineering
Electronic Technology
Department
Avda, Reina Mercedes s/n
41012 Seville (Spain)

Robert Denda
Enditel Endesa:
R&D Department
World Trade Center
Isla de la Cartuja s/n
41092Seville(Spain)

Samuel Muñoz
Enditel Endesa:
R&D Department
World Trade Center
Isla de la Cartuja s/n
41092Seville(Spain)

José Manuel Elena
School of Computer
Science and Engineering
Electronic Technology
Department
Avda, Reina Mercedes s/n
41012 Seville (Spain)

Joaquín Luque
School of Computer
Science and Engineering
Electronic Technology
Department
Avda, Reina Mercedes s/n
41012 Seville (Spain)

Abstract - **The aim of this paper is to show the utility of expert systems in the task of designing cable networks. At Enditel Endesa[1] we are currently developing a project called Datacab in collaboration with the Electronic Technology Department of the University of Seville. Datacab is an expert system for automatic routing and positioning of the elements of both the civil design and the cabling of a cable telecommunication network. It uses as input data from a GIS (Geographic Information System) and obtains an optimal design by applying a specific design rules database. Datacab's representation of the geographic information is entirely based on graphs. Among the advantages of the proposed expert system approach is the possibility to add new rules or to modify present rules easily if design criteria change.**

I. INTRODUCTION

A. Expert Systems

An expert system is a computer program, which uses non-numerical domain-specific knowledge in order to solve problems with a competence level comparable with that of a human expert who is able to efficiently solve domain-specific problems. Rule-based programming is one of the most commonly used techniques for developing expert systems [1]. In this programming paradigm, rules are used to represent heuristics, or "rules of thumb", which specify a set of actions to be performed for a given situation.

Expert systems have been successfully applied on telecommunication networks [2][3][4] and, in general, to the search of optimal solutions of problems for graphs [5]. Since a cable network's topology can be studied as a connect graph, it makes it perfectly suitable for applying such an expert system approach.

B. Cable networks

Current cable networks are typically implemented as HFC (Hybrid Fibre Coax) networks, where transmission media are optical fibres and coaxial cables, and potentially a parallel telephony network.

The cable network structure to be designed supports Cable television, Internet access and telephony services. The fiber optic network is structured by a head-end connected to a primary network, which contains a number of main nodes servicing between 30.000 to 90.000 homes each. These main nodes are connected to a network of several secondary nodes (secondary networks), each of which serves around 10.000 homes. In general, the secondary nodes also contain a RTC (Remote Telephony Center). The tertiary networks (distribution network) connect to the secondary node and service about 2.000 homes. Every tertiary network ends in ONTs (Optical Network Termination), which are connected to a Coax Network. This Coax Network finally distributes the signal to subscribers.

The Telephony Network typically starts at each RTC and has three levels. The first level connects each RTC to a ONT, the second level reaches each package of homes, and finally, the third level connects the subscribers' homes.

C. Datacab

Datacab is an expert system based on rules for automatic routing of a HFC telecommunication network as well as its corresponding civil design. Datacab will be applied to various cable communication networks with an average of more than 200.000 users each.

Some of the advantages of the proposed expert system approach are the possibility of adding new rules and to modify present rules easily if the design criterion changes. Design rules can be stored in a database in order to improve accessibility; finally the developed system can be used to train new staff.

II. DESIGNING DATACAB EXPERT SYSTEM

A. Knowledge acquisition

The first step in building an expert system is the knowledge acquisition. It involves eliciting, analyzing, and interpreting the knowledge about the current network design. Several techniques for extracting expert knowledge have been described in the literature [6]. We selected an approach of structured interviews with the human operators as the main acquisition technique. A structured interview is a conversation between the experts and the knowledge

[1] All intellectual property rights of the presented System are hold by Enditel Endesa. For further information, please contact any one of the authors.

0-7803-7474-6/02/$17.00 ©2002 IEEE

engineers. The difference between a structured interview and a normal meeting is that the knowledge engineer must decide on the questions and objectives to discuss before the meeting. A specialized technician was commended for carrying out the expert work. Along 5 sessions from 1 to 2 hours were necessary to build our first knowledge base. Available demonstrations about HFC system were additionally used.

B. Knowledge Representation

Once knowledge had been acquired, the next step was its representation. A class diagram and a rule-based system had to be built for it. The geographic elements with their location and re-design information (i.e., possible access to accommodations, previous channel locations, non-permitted façades...) are stored in the GIS application.

Class diagram

The class diagram of Datacab was designed with the help of the Rational Rose tool using an UML approach [7][8] and is composed of 7 packages, which are linked in the way shown in fig. 1. It includes the geographical elements (streets, buildings...), re-design information (buildings accesses...), coax elements (wires, amplifiers...) and civil elements (caskets, channels...) in addition to superclasses of all the previously mentioned elements (point-class, coax-class...). These superclasses were created to generalize some common properties of our objects such as distances in linear elements and signal losses in elements of the coax network.

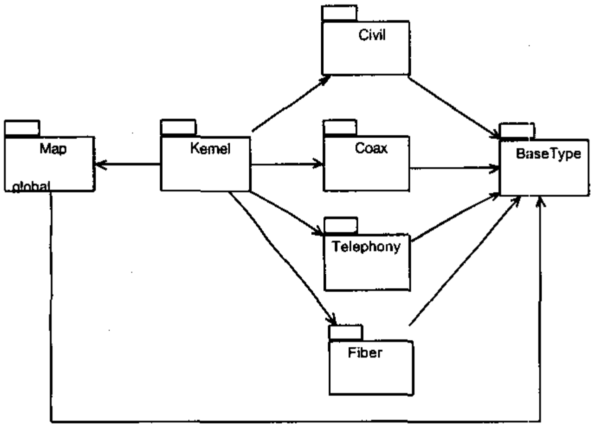

Fig. 1. Package diagram

The *Kernel* package is made up by the knowledge classes, e.g., the algorithm-class. All of Datacab's classes are derived from one superclass called *Gis* from the *BaseType* package (see fig. 2). This class embraces the main shapes of the geographical elements: point, segment and area. On the other hand, in the *Map* package are the classes related to city elements: buildings, blocks, streets, roads, parks and so forth.

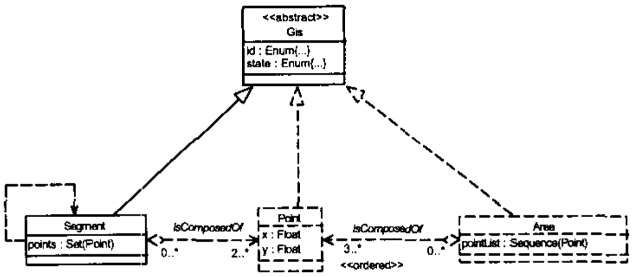

Fig. 2. BaseType package

The *Coax* package has a superclass called *Coax*, which contains three classes corresponding to the type of coax element: passive component (i.e., wires), active component (amplifiers) and power source (to feed active components). The different passive component classes of the coax design are shown in fig. 3. In this example (fig. 3), the advantages of the inheritance can be seen: inherited classes from *Passive* get the characteristics from coax elements and their geometric data.

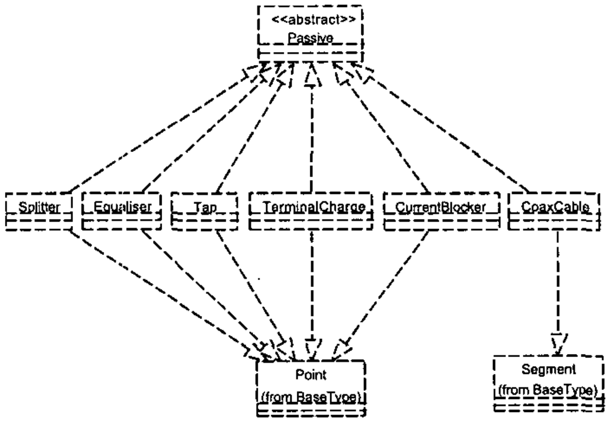

Fig. 3. Coax passive elements

Finally, the design of the civil component classes is made up by a main class called *Civil.* Similarly to the other superclasses, *Civil* has two child classes depending on their characteristics, distinguishing segments and points. The first class inherits from *Segment* whereas the second one from *Point.*

Datacab will instantiate one copy of each different child class of the coax diagram and civil diagram and the attributes with technical characteristics are filled in according to their constructor via database accesses.

Datacab's database contains all the characteristic data of the network elements as well as their subgroups used for the corresponding city or zone. The administrator only has to change the database for updating Datacab.

Rules-based system

Our knowledge base is a rules-based system: the knowledge is represented in the form of IF...THEN types rules (premises and conclusions), facts and assumptions about the problem. This reasoning system was chosen over others (e.g., case-based reasoning) due to the fact that the alternatives in the cable network design are subject to rules. So, when an expert carries out the design, he guides himself by priority rules and experience, with the main purpose of reaching every subscriber homes or future subscriber homes. Some rules in routing phases are shown in table I and the code of an example one in table II. Currently, our system contains about 80 rules.

TABLE I
SOME RULES FROM DATACAB

Static (not dependent of previously selected edges)		
Name	Priority	Subject
SESS-01	max	Cable over façade is better than cable via tubes.
SESS-02	mid	Cable via tubes under asphalt is better than under pavement.
SESS-03	min	The higher the savings in cable the better.
Dynamic (dependent of previously selected edges)		
Name	Priority	Subject
SEDC-01	max	No cycles in the graph of our cable network
SEDC-02	mid	Cable preferred if it does not cross avenue or Main Street.
SEDC-03	min	The lesser the number of adjacent edges the better

TABLE II
CODE OF SESS-01 RULE

```
(define-rule SESS-01
    (declare (salience 900))
    (object ?algorithm (instance-of algorithm500-1)
              (phase static-cost)
              (CS ?node-list)
              (RS ?non-node-list))
    (object ?street1
             (instance-of dc:street))
   (object ?street2
              (instance-of dc:non-façade))
        (test (not (equal ?street1 ?street2)))
        (test (not (member$ ?street1 ?node-list)))
        (test (not (member$ ?street1 ?non-node-list)))
        (test (superimposed ?street1 ?street2))
         =>
  (add-attribute-value ?algorithm CS ?street1))
```

III. DATACAB EXECUTION PROCESS

A. Phases

Datacab achieves the routing of the HFC network through different phases. The different phases of Datacab's execution are shown in fig. 4. For coding of these phases we used a powerful tool for expert systems called Art*Enterprise.

The first and the last process of the execution diagram (GIS Interpreter) are relatively simple due to the ease of the GIS tool to export data to a desired format. As a result, the interpreter module generates a file with the GIS information facilitating the conversion process.

The attributes of the geographical elements are specified in our class diagram and the objects created in the second step are instances of these classes.

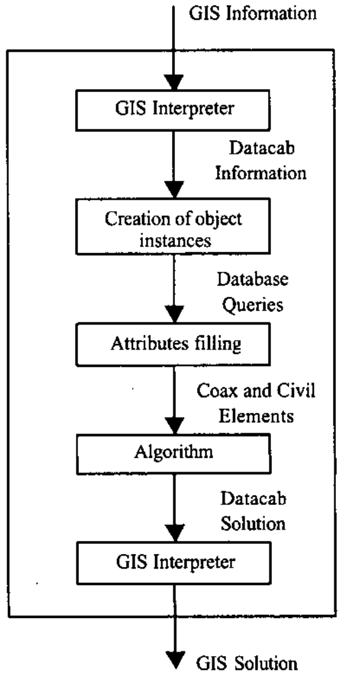

Fig. 4. Execution diagram of Datacab

B. Routing algorithm

The *Kernel* package (see fig. 1) has embedded the *Kernel* superclass and *Algorithm* class. These classes, which make up the knowledge base, take care of firing the rules in every phase of the algorithm. Besides, the results are stored in the class *Algorithm.*

The kernel algorithm, currently under development, will be made up of phases each one of which will fire a group of rules. The first design idea of the algorithm consists of first performing the cable network routing, and then locating thee coax and civil elements. In the case that it turns out to be impossible to place the elements within one single cable network routing, it will be carried out during a second routing.

The first stage of the search algorithm will be based on the Kruskal's algorithm [9], which is a historical minimum expansion tree search algorithm. These algorithms have a connected graph as input and generate a tree covering the whole graph with the lowest possible cost. In particular, Kruskal's algorithm selects the lowest cost edge in each one of the iterations, as long as this does not generate any cycles.

The difference between Kruskal's algorithm and our algorithm is in the dynamic characteristic of the latter one. While the edge cost in Kruskal's algorithm does not depend on the previously selected edges, in our algorithm the cost will be allowed to vary. The different phases of the algorithm of Datacab are shown in fig. 5.

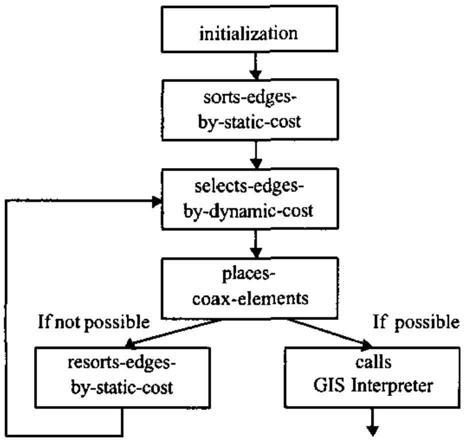
initialization
sorts-edges-by-static-cost
selects-edges-by-dynamic-cost
places-coax-elements
If not possible
If possible
resorts-edges-by-static-cost
calls GIS Interpreter

Fig. 5. Datacab algorithm phases

In addition, we are researching another alternative for the algorithm. It consists in placing coax and civil elements at the same time while the cable network is being designed. This solution is more in accordance with the working mode of a design expert but it has been postponed due to its more complicated implementation.

The features of the coax elements will be passed through a database access and depend on the city or zone. In this way, Datacab will release the signal calculations using the obtained parameters with the purpose of assuring the required signal levels in every point of the routing. Once the kernel finds a solution, this solution becomes available in the GIS database, which allows its graphical representation (fig. 6).

IV. CONCLUSIONS

Up to now, we have developed a first prototype of the expert system for the automatic design of cable networks and placing of civil elements. The initialization phase and the two phases of the network routing have been implemented. We are currently developing the stage of the project that will allow us to place the coax and civil elements according to the established rules and signal level calculations.

Besides, we are preparing a test method (validation process). Once finalized these phases, an expert will verify the correct network designs carried out by Datacab (verification process).

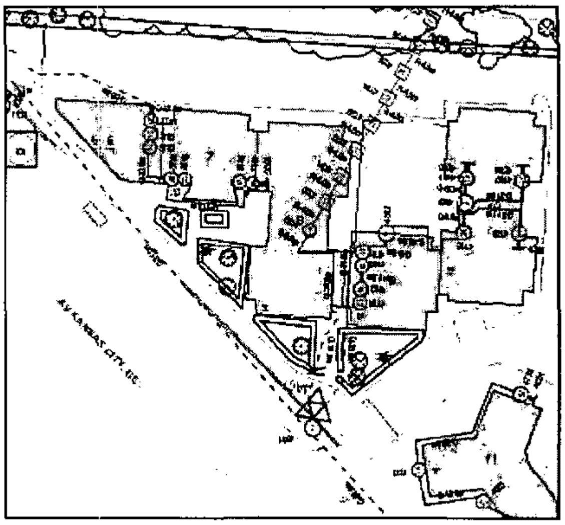

Fig. 6. Example of routing obtained for a cable network

V. REFERENCES

[1] E.M. Awad. "Building Expert Systems". West Publishing Company, 1996.

[2] C. León, M. Mejías, J. Luque y F. Gonzalo. "Expert system for the integrated management of a power utility's communication system". *IEEE Transactions on Power Delivery*, vol. 14, n° 4, pp, 1208-1212, Oct. 1999.

[3] Jay Liebowitz ed. *"The handbook of Applied Expert Systems"*. CRC Press, 1998.

[4] I. Vlahuvas et al. "ExpertNet: An Intelligent Multiagent System for WAN Management". IEEE Intelligent Systems, vol. 17 , n°4 , pp, 62-72, 2002.

[5] E. Castillo, J. Gutierrez, Ali.S Hadi. "Sistemas Expertos y Modelos de Redes Probabilísticas". Monografías de la Academia Española de Ingeniería, Madrid, 1998.

[6] K.L. McGraw and K. Harbison-Briggs. "Knowledge acquisition: Principles and guidelines". Prentice-Hall, Englewood Cliffs, N.J.

[7] G. Booch, J. Rumbaugh and I. Jacobson. "UML. El Lenguaje Unificado de Modelado", Addison Wesley, 1999.

[8] W. Boggs, M. Boggs. "Mastering UML with Rational Rose", Sybex Inc, 1999.

[9] R. Sedgewick: "Algoritmos en C++". Addison Wesley Iberoamerica, 1995.