PROJECT MANAGEMENT USING GRAPHICS

Fabio Pettinati

Electrical Engineering Department
Politechnic School
University of Sao Paulo
Sao Paulo - Brazil

ABSTRACT

This paper addresses the problem of lack of visualization most managers face when using computer-based project control systems: although highly relevant information is generated, usually no graphical output is produced. This paper presents a system called UniPert that automatically produces high quality drawings showing all activities present in a project and the relationship among them. A detailed description of UniPert's major components and algorithms is presented as well as examples of its actual use. The main contribution of this paper is the integration of many different techniques and concepts that led to the development of the UniPert system.

RÉSUMÉ

Cet article se rapporte au problème du manque de visualization que la plupart des entrepeneurs rencontrent lors de l'usage de systèmes de controle de projet assisté par ordinateur: bien que des informations de haute importance sont fournies, normalment aucunne sortie graphique est prévue. Cet article présente le système denommé UniPert qui produit automatiquement des diagrammes indiquant toutes les activités prenant part d'un projet et leurs correspondantes relations. Nous presentons ainsi une description détaillée de l'UniPert suivant ses principaux eléments et respectifs algorithmes, de même que quelques examples de son emploi réel. La principale contribution de cet article consiste de l'intégration de plusieurs theories et techniques qui menèrent au développement du système UniPert.

KEYWORDS: business graphics, project management, parsing, data structures.

INTRODUCTION

The last twenty years have seen a tremendous growth in the use of computer-based project control systems. The increased availability of such systems is mainly due to the following facts:

- The expanding familiarity of project management personnel with this tool;
- The pressure to complete work quickly before inflation takes a large share of the budget;
- Management hope that an investment in computerized project tracking will shorten project time and decrease costs.

A project control system consists of computer programs that analyze the activities present in a project, as well as the topological relationship among them. Inside the computer, a project is represented as a network composed of activities that have to be completed and a set of relationships that have to be observed.

The two most common types of network notations are:

- ACTIVITY ON NODE or PRECEDENCE NOTATION

  In this notation, each activity is considered to be a node in the network and relationships between two activities are called precedences and represented as edges connecting the nodes associated to the activities. Different types of precedences are allowed and, for each type, a minimum time span (referred to as a lag) must separate the beginning or the completion of both activities (referred to as predecessor and successor).

- ACTIVITY ON ARROW or ARROW DIAGRAM

  In this notation, each activity is designated in the network by an edge (arrow) that starts at a predecessor node and ends at a successor node; both nodes are called events and have time values associated with them.

Project control systems using any of the above notations will calculate start and finish dates for all activities, indicate how long each activity can be delayed (referred to as float) without delaying the whole project and highlight the set of activities with critical float values.

In spite of the power and extended usage of project control systems, even today project management is sometimes considered to be an art which involves skills for dealing with a great number of concurrent and resource sharing activities. This is due to the fact that those involved in this sort of activity often rely on their own previous experience and intuition rather than on objective conclusions.

Part of the problem is caused by the project control systems available in the market today. In spite of being widely used, these systems usually provide users with just lengthy user-specified printed reports. To overcome this lack of visualization and also to enhance comprehension, several companies started offering systems that provided users with graphical output more suited to their needs. Another solution that is also available in the market today acts as a post-processor to existent project control systems and generates the desired drawings.

Both approaches were not readily available in Brazil in 1982, a time when several companies started looking for this kind of graphical output. Due to the restrictions posed by the Brazilian government for companies to import software, it was felt that a locally developed system acting as a post-processor to available project control systems should be the solution. That was the start of the research that led to the development of the UniPert system.

THE UNIPERT SYSTEM

The UniPert system comprises a control program that accepts user input parameters, reads in project data and produces the desired drawings. UniPert can be interfaced to project control systems using the precedence notation for networks and will produce both precedence diagrams with activities sorted in topological or chronological order and Gannt barcharts.

UniPert was developed with the following objectives in mind:

- Drawings produced by the system should have the same quality as those produced by draftsmen;

- System portability and graphics device independence should be enforced;

- The system should expend as little computer resources as possible;

- The system should be user oriented, requiring neither extensive training nor excessive data input.

These objectives were attained by designing UniPert as shown in (Fig. 1). As it can be seen, UniPert can be divided into five major components or subsystems, each one comprising several routines:

- CONTROL PROGRAM

  This subsystem coordinates the proper operation of UniPert's remaining subsystems while ensuring correct data flow.

- DATA ACQUISITION

  This subsystem is responsible for interfacing UniPert to various project control systems such as PROJACS, CIPREC or MAPPS. This interface consists of a file produced by the project control system containing all activities and precedence information present in the network used for project analysis.

- COMMAND INTERPRETER

  This subsystem is responsible for reading and parsing all free-form commands provided by users. These commands are keyword oriented, can be abbreviated or written in full and their syntax is extremely flexible. Parsing is accomplished by a non-recursive top-down command syntax analysis, error reporting and by calling action routines associated with the commands.

- GRAPHICAL OUTPUT

  This subsystem is responsible for drawing all charts produced by UniPert and it consists of a suite of routines patterned after the GKS standard (strict adherence to GKS is not enforced, but all important concepts and functions are present). Although the list of available device drivers includes both raster terminals and pen plotters, only the latter are used by

UniPert; the reason is that users usually take copies of the drawings for distribution and further analysis.

- ALGORITHMS AND DATA STRUCTURES

  The algorithms and data structures employed play a vital role in UniPert: both are responsible for enhancing system overall performance. From holding project data to positioning activities and routing precedences accross the network, these are only a few examples of algorithm and data structure usage in UniPert.

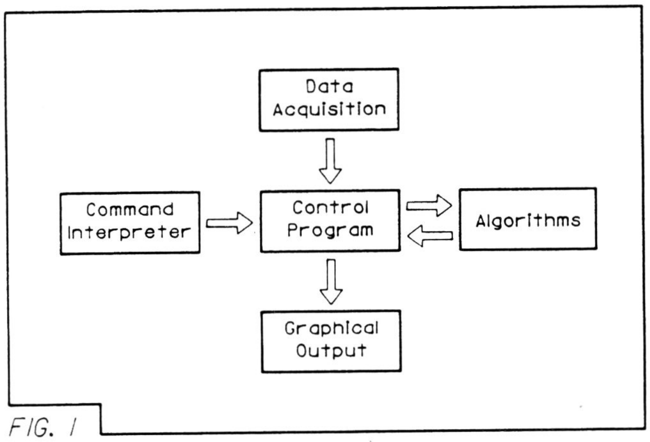


FIG. 1

Almost all UniPert's routines were coded in BLOKFOR, a structured language based upon FORTRAN-77; the only exceptions to this rule are some low level routines for address manipulation that in some machines (e.g. IBM) have to be coded in assembly language.

The BLOKFOR language can itself be translated into machine code by a two-pass procedure: the first pass generates an intermediate program by translating each BLOKFOR statement into equivalent FORTRAN-77 statement(s) and the second pass inputs this intermediate program to the FORTRAN-77 compiler.

UniPert can be run on computers providing virtual memory allocation that can be directly controlled by an application program. At present there are two UniPert releases running on IBM computers under OS/VS1, MVS and CMS and on VAX computers under VMS. Although there are some minor differences between both versions, source code management is easily handled by the BLOKFOR compiler itself.

The following sections will provide readers with aditional information about the preceding subsystems.

THE DATA ACQUISITION SUBSYSTEM

UniPert can be interfaced to various project control systems by means of a file containing selected activities and precedences comprising a network.

Although the format of this interface file may change according to the system in use, its structure must remain the same: activities have to be sorted in lexicographic order and after each activity there must be a list (possibly empty) of its precedences. Besides activity and precedence data, an interface file contains several header records that allow UniPert to identify the incoming network, its work breakdown structure or run date; this general information is used afterwards to properly place legends in the charts to be drawn.

The format of the interface file to be used by UniPert can be found in a master file which describes the attributes associated with activities and precedences. For activities, the minimum set of attributes includes identification code, float and a start date; for precedences, the minimum set includes preceding activity code, precedence type and lag. In addition to a flexible interface file format, users may specify which information will be displayed in the charts produced by UniPert; this option allows users to effectively taylor UniPert's output to suit their needs.

Since interface files may contain several networks, each one with an unknown number of activities and precedences, data acquisition is performed by a two-pass procedure repeated for each network present: the first pass writes the next network in the interface file into an intermediate file and allocates enough virtual memory to hold all activities and precedences; the second pass completes the process by reading the intermediate file and storing activities and precedences into the virtual memory data structures.

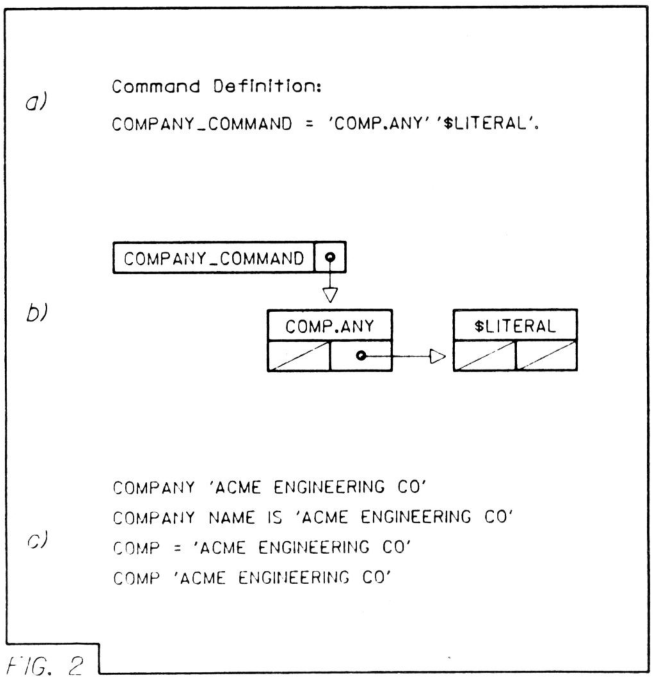
THE COMMAND INTERPRETER

The command interpreter used by UniPert can be classified as a 'one-symbol-lookahead without backtracking top-down parser' [WIRT76], [WIRT77]. The parser is similar to 'a recursive-descent parser except that recursion is removed by using an internal stack. Grammars accepted by the command interpreter are of class LL(1) [LEWI76] and they are represented inside the parser as a syntax graph. This syntax graph consists of nodes that correspond to the terminal and non-terminal symbols present in the grammar to be parsed.

Command parsing is performed by traversing the syntax graph in response to the tokens recognized by a scanner. The scanner recognizes identifiers, keywords, literals, numbers (both integer and real), symbols, indirect command files [DED080] and passes these tokens to the parser that will compare them to the syntax graph node contents. Graph traverse is aided by a stack which holds pointers to the nodes that have yet to be recognized. A pointer to the successor of a node is pushed-down into the stack whenever this node corresponds to a non-terminal symbol. Upon a non-terminal symbol identification, a pointer is popped-up from the stack and the analysis proceeds with the pointed node.

Every node in the syntax graph has attached to it the number of a 'semantic' routine that corresponds to the action one has to take upon node recognition; this scheme provides great flexibility since it allows for a complete separation of syntax and semantics [SETZ79].

To make interaction with UniPert easier, the command interpreter allows users to input commands with keywords in abbreviated form or with extra words to improve readability. (Fig. 2) gives an example of a command used by UniPert to identify the company which is running the system; (a) shows the command definition, (b) shows how the command is represented in the syntax graph and (c) shows some alternate forms of its use.


a) Command Definition:

```
COMPANY_COMMAND = 'COMP.ANY' '$LITERAL'.
```

c)

```
COMPANY 'ACME ENGINEERING CO'
COMPANY NAME IS 'ACME ENGINEERING CO'
COMP = 'ACME ENGINEERING CO'
COMP 'ACME ENGINEERING CO'
```

FIG. 2

GRAPHICAL OUTPUT PRODUCED BY UNIPERT

UniPert produces two types of graphical output: precedence diagrams with activities sorted in topological (Fig. 3) or chronological (Fig. 4) order and Gannt barcharts.

The main difference between both precedence diagrams is how the activities are placed: in the topological order, activities are sorted so that each activity is placed to the right of all its preceding activities; in the chronological order, activities are sorted so that all activities to the left have lower start date values than those to the right. Another way of showing the difference between both placement methods is to notice that in topological order precedence flow is always to the right; in chronological order some precedences may turn back, thus flowing to the left.

Gannt barcharts show no precedence information at all but clearly indicate how long each activity will take to be completed or how fast is work being done. Since drawing barcharts do not pose many challenges to UniPert, this paper will concentrate just upon precedence diagrams.

When drawing precedence diagrams, activity placement plays a trully important role: it determines how aesthetic the final chart will be and how long it will take to route precedences through the set of rows and columns activities had been placed into. As it can be seen in (Fig. 5), activities are placed as boxes into the rows and columns of a matrix. The distance between rows is constant, allowing up to six horizontal precedence segments to be routed through. The distance between columns, on the contrary, is not fixed and varies according to the number of activities placed into them; this scheme garantees that no two vertical precedence segments will overlap each other.

Associated with every activity, there are two points called leading (to the left) and trailing (to the right) connectors. These points collect all precedences so that arrivals are always made at the left side and departures are always made from the right side. This rule works perfectly for activities sorted in topological order, since preceding activities are placed to the left. For chronologically sorted activities, there may be some precedences in which the predecessor activity comes to the right of the successor one; in this case, the reversed precedence is drawn as a dashed line (Fig. 6).

Project management allows up to four different types of precedences:

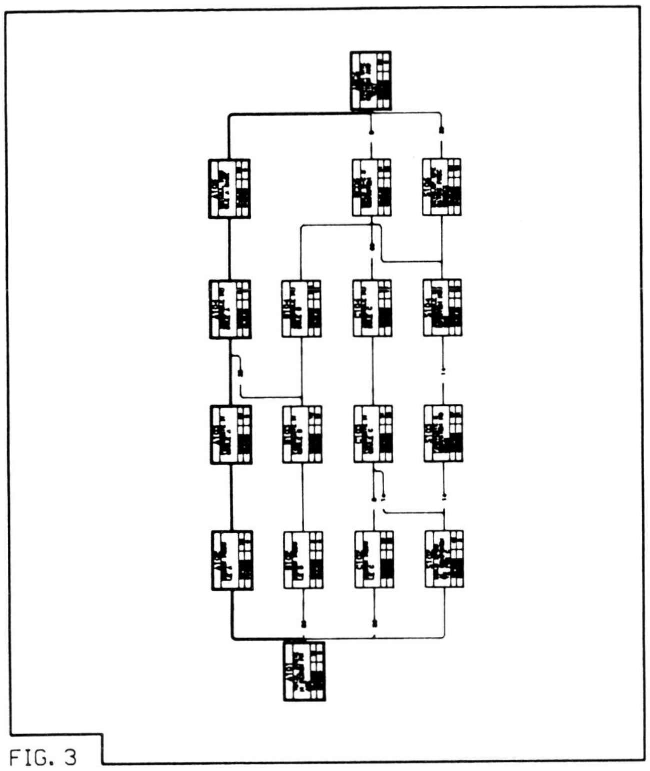


FIG. 3

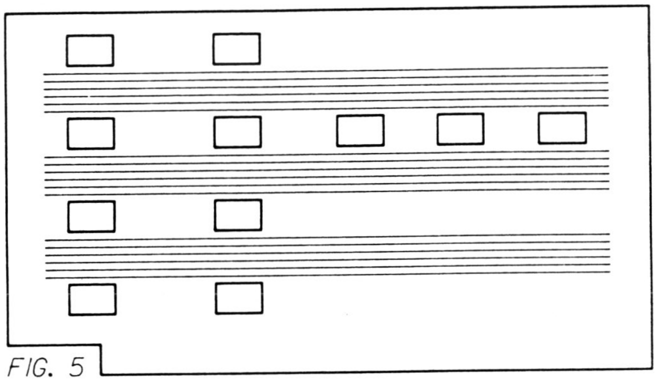


FIG. 5

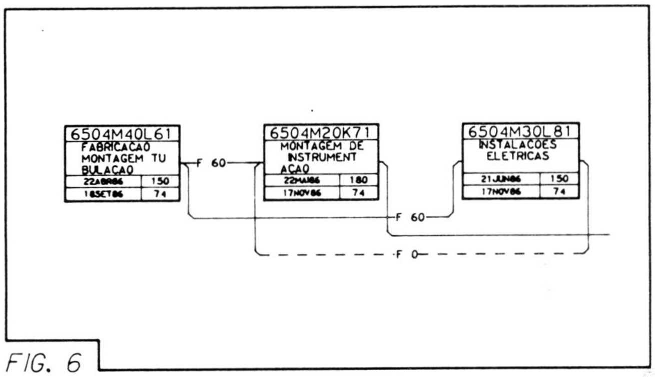


FIG. 6

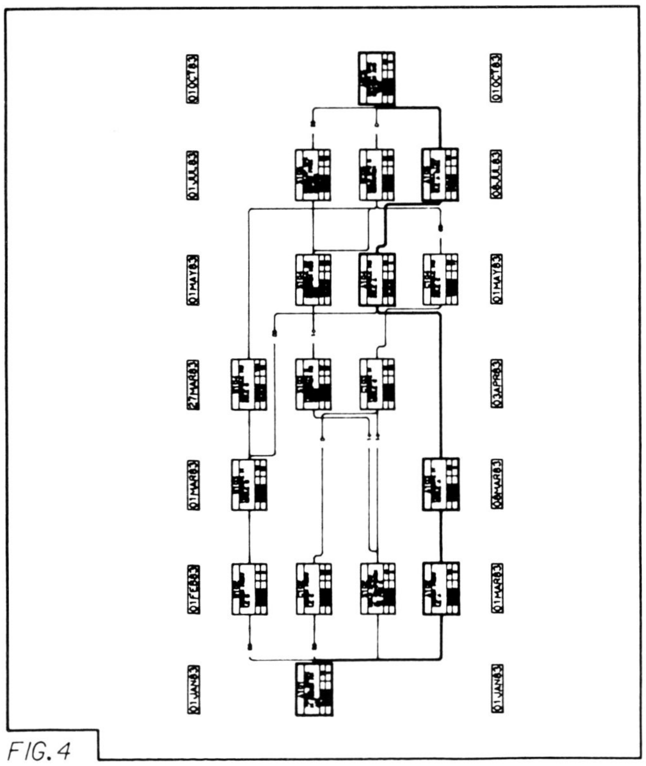


FIG. 4

- FINISH TO START (FS) where the beginning of an activity depends on the completion of its predecessor;
- START TO START (SS) where the beginning of an activity depends on the beginning of its predecessor;
- FINISH TO FINISH (FF) where the completion of an activity depends on the completion of its predecessor;
- START TO FINISH (SF) where the completion of an activity depends on the beginning of its predecessors.

To make clear which type of precedence is connecting two activities, UniPert draws along the precedence's horizontal segment its type and lag value. UniPert always draws this information between activity columns in a position where no precedence will ever cross it (Fig. 7).

As it was stated before, users can specify which information will be posted inside activity boxes. Up to fifty user-specified formats can be accomodated and UniPert provides a default format that includes: activity code and description, early start and finish dates, duration, float and a milestone flag (Fig. 8).

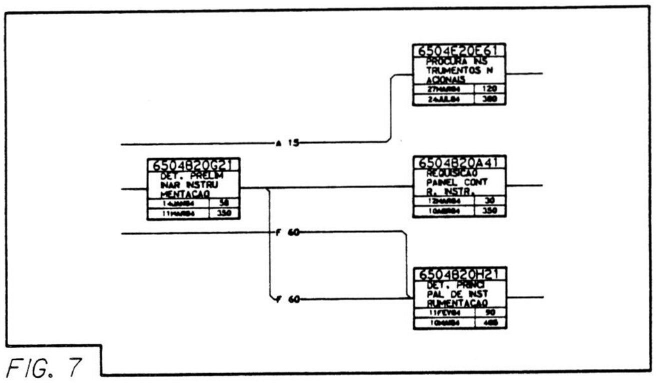

6504E20E61
PROCURA INS TRUMENTOS N ACIONAIS
6504B20G21
DET. PRELIM INAR INSTRU MENTACAO
6504B20A41
REQUISICAO PAINEL CONT R. INSTR.
6504B20H21
DET. PRINCI PAL DE INST RUMENTACAO
A 15
F 60
F 60

FIG. 7

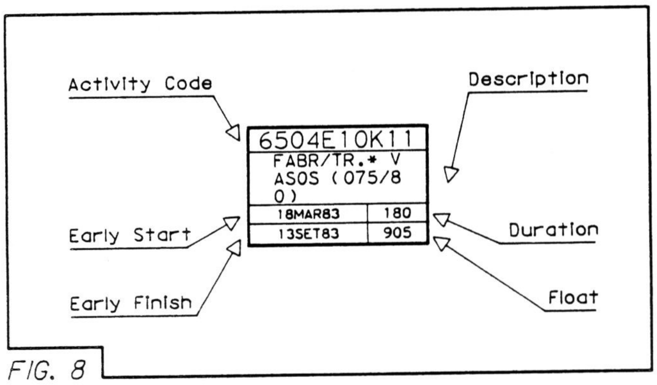


FIG. 8

Activities with critical float values are properly highlighted so as to reflect their importance. Highlighting is accomplished by drawing critical activities and precedences with a different linewidth (Fig. 9).

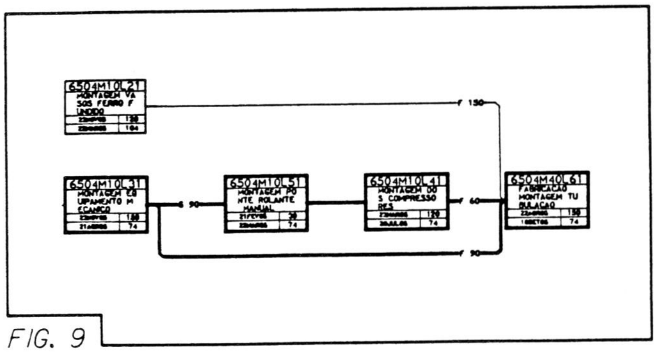


FIG. 9

All charts produced by UniPert are framed inside four different but standard page sizes, ranging from A0 (118.8x84.0cm) to A3 (42.0x29.7cm). When charts are divided into pages, there are some precedences which cross page boundaries. UniPert interrupts each crossing precedence and places at interrupt points arrows indicating flow direction and activity codes giving 'from-to' information (Fig. 10).

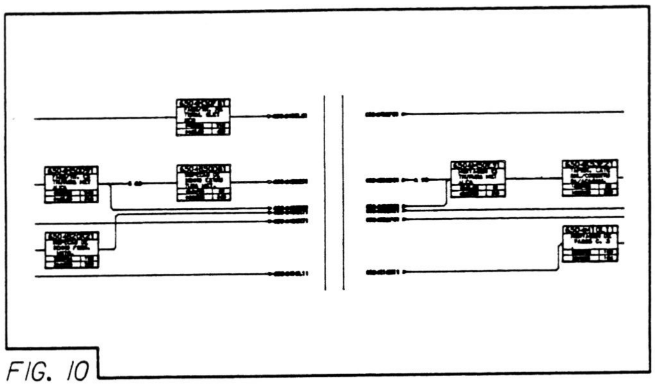


FIG. 10

Each page is identified by a legend placed at the lower-right corner; this legend gives the name of the company using UniPert, displays the structure of the network submitted, run date and page number and also identifies the fields posted inside activity boxes (Fig. 11).

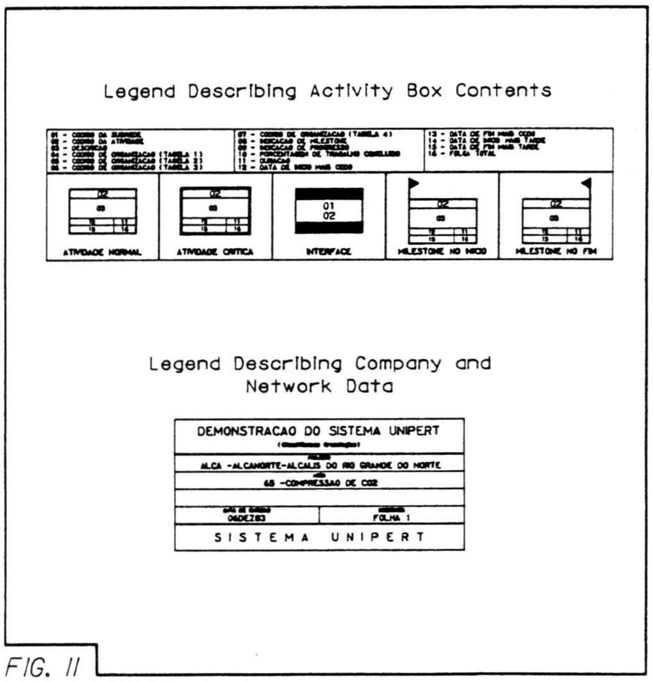

Legend Describing Activity Box Contents
ATIVIDADE NORMAL
ATIVIDADE CRITICA
INTERFACE
MILESTONE NO INICIO
MILESTONE NO FIM
Legend Describing Company and Network Data
DEMONSTRACAO DO SISTEMA UNIPERT
ALCA - ALCANORTE-ALCALIS DO RIO GRANDE DO NORTE
6B - COMPRESSAO DE CO2
06DEZ83
FOLHA 1
S I S T E M A   U N I P E R T

FIG. 11

ALGORITHMS AND DATA STRUCTURES

The main data structures used by UniPert comprise a table for storing activities, a queue to be used for sorting activities in topological order and several linked lists for storing precedences and also for precedence routing. All these data structures are considered to be dynamic since they only exist in virtual memory at run time.

As it was seen before, activities are placed as boxes into the rows and columns of a matrix. Column placing is accomplished by first sorting activities in topological or chronological order; this new ordering will be used afterwards to assign activities to the columns in a left-to-right process. It may happen that some rows have no enough space to accomodate all assigned activities; in this case, the extra activities are transferred to next columns until all columns are properly filled.

Row assignment is accomplished by distributing activities so that for each activity the sum of the vertical distances to all preceding activities is minimized. Activities sharing the same predecessors will be placed into the same row; to avoid this collision, activities are evenly spread around their assigned rows.

Precedences are routed through channels placed like a grid over the drawing; grid dimension is set to 0.25cm (about 1/10in) and users have no control about it. To easy the routing process, activity boxes with leading and trailing connectors are transformed into channel obstacles (Fig. 12); using this approach, the router's only task is to find pathes that avoid channel obstacles and join activity connectors.

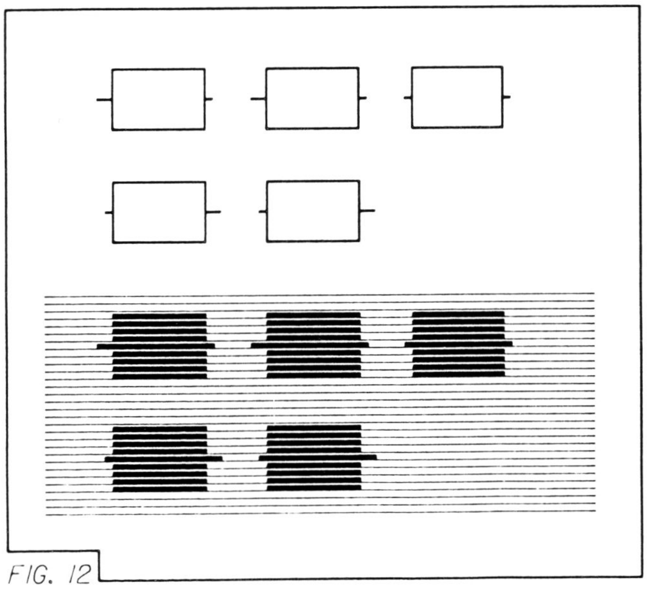


FIG. 12

Path finding is accomplished by extending two vertical lines through both connectors and by searching for an horizontal segment that goes from one line to the other without hitting any channel obstacles (Fig. 13). If such segment is ever found, it is inserted into the list of channel obstacles with a tag identifying it as being part of a precedence.

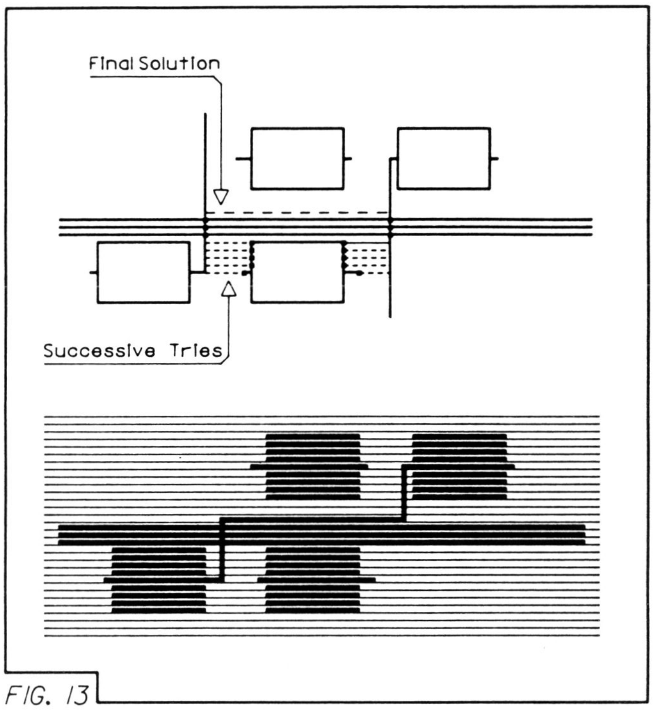


FIG. 13

After the routing process finishes, tagged channel obstacles and their associated vertical lines are drawn thus completing the algorithm. Precedences not routed by this procedure are printed in a report and left to be completed by hand; except for just one extremely dense network, where it attained a score of 97%, UniPert never failed to route all precedences.

The algorithm used for precedence routing is very efficient and although routing time accounts for up to 40% of total running time, overall performance is very high. A network generated by PROJACS consisting of 77 activities, 110 precedences took 23s of CPU time to be completed in an IBM 4341 running under MVS; running time was observed to be roughly proportional to both the number of activities and precedences.

EXAMPLES OF UNIPERT'S USAGE

UniPert has been used for the last two years by several companies as an aid to solve their project management problems. Among these companies one can find:

- A major civil construction company that used UniPert to manage the building of an international airport in Brazil;
- A state railroad company that used UniPert to manage its expansion program;

- An engineering company that used UniPert to manage the construction of a chemical processing plant.

Almost all companies employing UniPert use PROJACS as their project control system, one company uses CIPREC and another one will start using MAPPS.

CONCLUDING REMARKS

This paper presented the UniPert system for drawing charts that complement the output produced by current project control systems. The precedence diagrams and Gannt barcharts produced by UniPert enhance project comprehension and provide users with a new tool for project analysis.

Looking at the software modules that comprise UniPert, one can see the fusion of several computer science techniques that led to the production of a system that is not only modular in nature but also easy to use, understand and modify.

BIBLIOGRAPHY

REFERENCES MENTIONED IN THE TEXT

[DEDO80] Scanner Design
Dedourek, J. M.; Gujar, U. G.
Software-Practice and Experience
Vol. 10, 959-972 (1980)

[LEWI76] Compiler Design Theory
Lewis II, P. M.; et al
Addison-Wesley Publishing Co.
1976

[SETZ79] Non-recursive Top-down Syntax Analysis
Setzer, V. W.
Software-Practice and Experience
Vol. 9, 237-245 (1979)

[WIRT76] Algorithms + Data Structures = Programs
Wirth, N.
Prentice-Hall
1976

[WIRT77] Compilerbau (in german)
Wirth, N.
B. G. Teubner (ed)
1977

REFERENCES NOT MENTIONED IN THE TEXT

PROJACS Reference Manual
CIPREC Reference Manual available from:
International Business Machines Co. (IBM)

MAPPS Reference Manual, available from:
Structural Programming, Inc.
83 Boston Post Road
Sudbury, MA 01776
U.S.A.

E-Z-PERT Reference Manual available from:
Systonetics, Inc.
801 E. Chapman Avenue
Fullerton, CA 92631
U.S.A.

UniPert Reference Manual available from:
Tecninger Sistemas Ltda
Av. Rouxinol,200 Cj71
Sao Paulo 04516
BRAZIL