AN AREAL DATA MANAGEMENT PACKAGE

UGO SCHIAVONI

DIP. DI TECNICHE DELL'EDILIZIA E DEL CONTROLLO AMBIENTALE - UNIVERSITA' DI ROMA "LA SAPIENZA"

VIA EUDOSSIANA.18 ROMA

1.

The Areal Data Management Package was created to fulfill the need for a data manipulation system on the basis of a grid cell data structure. The package was originally developed for use by research operators in land planning and natural resources [1].

Over the past three years the package has been used extensively by various users, including students under and postgraduates.

The ADM has been designed for users having no experience with computers, but it does assume understanding of resources and land planning information.

The specific manipulative capabilities of ADM Package are designed to help land use planners analyze the natural and man-made characteristics of an area. The ADM is intended as a tool to manage spatially disposed thematic and categorical information, in many cases supplementing or replacing more conventional methods of analysis.

However, ADM does not create criteria, but only "reshape" information according to user's specifications. It is not an analytical model, pre-programmed to solve a specific problem; users are requested to define personally the analytical procedures in order to solve problems and to get specific tasks.

The source program of ADM is written in basic. The package, implemented on Apple// computers with 48k and semi-graphic printers [2], is completely interactive; every function is enabled by the user's answers to program's demands. In the standard configuration the package has functions which provide store, manipulate and display or print mapped thematic data. Specific functions and subroutines, developed occasionally by users-programmers, have been chained to the package.

2.

The data sources for ADM are traditional thematic maps; such maps describe the state of an area or region identifying geographic zones, to each one is normally associated a "description" of the phenomena under analysis. The kind of "description" depends by the type of measurement scale used; there are four scales: nominal, ordinal, interval. and ratio scales.

In order to manage different thematic maps, ADM uses grid cell data structures to record these maps in a digital format. Operationally, an abstract grid is defined and superimposed on each thematic map; the grid cells are the elementary units on which the data are collected.

In the ADM, is used a square cell grid. Operations issued do not request specifications about real dimension of the cells; the dimension is stated by user, defining a) the metric scale of thematic maps, and b) the size of superimposed grid quadrates. Each user can obtain the desired detail by choosing a combination of a) and b). The square shape of cells is actually referred only to print subroutines in the package; other subroutines can operate on rectangular grids.

The extent of grid on which programs operate is strictly fixed in 1600 cells. arranged on forty rows, each one containing forty cells; all files recording stored information are operated with reference to this dimension; no program can write or read a smaller part on the discretion of user. This extent of grid - forty rows by forty columns of cells - is conventionally named *table*; programs require a fixed numbering form of tables.

The package assumes that users could need more than one table to cover area under analysis, at required scale and appropriate grid size; In this respect, users are allowed to work with a maximum of 16 contiguous tables, at their turn arranged on four rows and four columns. Tables are identified in this set by a two figures numbering, in accord to their position in the set - row number and column number.

Cell position in the table is identified in two different ways: some programs use a progressive numbering - 1 to 1600 - from upper left end corner to the lower right corner; others use a cartesian co-ordinates system with the origin in the upper left corner.

Users would store and manipulate informations on more than only one phenomena. land use, vegetation, soils, etc.; in other words, to analyse an area user could require to store many different thematic maps introducing for each cell codes describing phenomena values in that location. ADM meets this need allowing the storage of the same table everytime is required by users, marking each storage by the "name" of phenomena described in it, conventionally named "variable".

Variable's names are completely free - but cannot be longer than 12 characters - and are managed by a specific subroutine. There is a third marking element in the names of tables, named "version number"; it will be useful when user will create a new table recoding an original stored table.

Codes for information storage are only numerical; all programs read alphabetical codes as zeros. However, the numerical shape of codes can fulfils needs for both nominal and ordinal classification. Code's length is free, but it is hardly advisable to use the same length for codes in the same table. in order to facilitate the interpretation of analysis.

Original table's names are very explicit in the presentation of markers - for instance, TAV.34 VAR.ALTIMETRIA/0 - ; files resulting from original files manipulation have less explicit names, although they contains all three markers described (table number, variable name, version number).

3.

In Figure 1. are shown relations between package's programs, between programs and data files, between data files and some external library programs.

Programs are grouped into five functional blocks; the VARSET program in block one manages "variables" setting and access to all other programs. In every work session programs are booted from VARSET, to which session control comes back after each program execution.

Other programs are not directly connected, except graphic printing programs, which at their turn are managed by STAGRAF MAIN program (Figure 2.). The indirect program connection does not allow sequential execution of different programs; therefore the standard operational sequence will be:

a) setting variables in VARSET

b) booting the required program

c) retrieving stored data from required program

d) processing data in required program

e) storing new data, or printing data

f) come back to VARSET

By VARSET user can create a maximum of ten variables sets, each one containing ten variables; variables sets are stored in different files. among which user can choose the set he intends to use during the session. Every program called in execution by VARSET catalog, will begin execution reading and displaying required variable set; user will choose a single variable, table and version number, creating the data file name that will be retrieved and processed.

The described variable managing system is intended to free the package from restrictions on subject of analysis, allowing use by anyone involved in processing locational data.

Programs in block two (Figure l.) accomplish functions of inputting, storing and correcting data files; they differ in functions fulfilled and, mostly, in interaction with user. By CARSER user can input data series into 1600 cells tables; CARSER requires user to input data in a sequential order - from left upper corner to right lower - going on in filling of table row by row. By CARSER user can also re-write codes in any cell he wants, correcting tables previously stored; cells have a progressive numbering from 1 to 1600.

CARPUN too allows inputting and correcting, but it interacts more directly with user; he can move a cursor trough the 1600 table cells displayed on screen writing in codes, or correcting existing values. There is no restriction in input sequences; cells values are displayed on screen with alphanumeric characters; four status rows give information on cursor position - both in progressive numbering and cartesian coordinates -, current cell value, etc.

By CORR program user can only correct existing tables, in the same way he can do by CARSER.

In addition to tables data files. CARSER and CARPUN can create an other kind of files, in which are listed cells external to the study area in every single table. Those cells will be automatically excluded from input operations, shortening input and making user sure of different variables boundary collimation.

TRASL, ESPANS, RIAGR programs allow respectively a) creation of tables shifted from standard position in the 16 tables set, b) expansion of a table in 16 new tables - and of each cell in 16 cells, changing scale but not information quantity - , c) aggregation of 16 tables in one table syntethizing 16 cells in one. In all three programs new tables are created from existing previously stored tables; table numbering will be modified from two to four or six figures to identify various tables type.

INTOR program assigns value to each cell according to its relative proximity to the locations of a specified condition (such as highways, urban centers, etc.). Each search origin condition generates a series of radiating proximity zones increasing in value with distance from specified origins.

COMP and CONF programs, in fourth block (Figure 1.), have the main task of computing frequencies in cells values on one (COMP) and two (CONF) tables. COMP provides listing of cells values in a table, computing frequence of each value by number of cells. CONF works on a couple of tables - same set position but different variables - computing frequence of value couples founded in the 1600 couples of corresponding cells. From both program computing results can be stored in files retrievable by library programs for further processing; immediate print outputs are available as listing, bar diagrams, matrix (Figures 3.,4.,5.).

An other important function accomplished by COMP and CONF is recoding; after frequence computing user can modify original cells values, creating a new table that will be stored without deleting original tables. Recoding is an exclusively nominal changing of codes: it only allows substitution of a code (COMP) or a couple of codes (CONF) with new codes. In the first case, user can perform a grouping of original values, reducing them in number; in the second case new codes are meaningful of complex cells situations. New tables are stored with same original names and new version number (COMP) and with a file name synthesizing all original file names (CONF).

Both programs can be used repeatedly; in their use is required the maximum user effort in clearing tasks and meanings of recoding. To avoid loss of recoding, a correspondence list of original and new codes is automatically stored after each new table creation.

Lower asterisked block in Figure 1. includes printing programs; there are numerical, alphanumerical and graphic printing programs.

STAMP program manages numerical printing, producing a square matrix - forty by forty -where can be printed a maximum of two figures of cells codes for each cell position; length codes exceeding, user is allowed choosing which figure couples to print (Figure 6.).

STAMPAN program runs in the same way, replacing numerical codes with alphanumeric characters, whose choice is at user's discretion; program does not supply a code list of table, therefore the availability of numerical stamp or frequence computing list is highly advisable in defining association numerical codes - alphanumeric characters. However, program does not require user to indicate a character for each numeric code: it prints blanks for ignored codes (Figure 7).

There are three basic graphic printing programs (STAGRAF 1, STAGRAF Z, STAGRAF 16); they all produce images on monitor by reading data files, printing and/or saving them on disk according to user's options. A fourth program (STAGRAF) can only retrieve stored images and print them.

STAU'RAF 1 works on one table time by time, drawing on screen 4x4 pixels 5ymbols associated to cells codes; user can select symbols in a catalog of twenty, which can be matched for more complex effects. Each table is then displayed as a square of 160X160 pixels, whose printing i5 executed by "dumping" (Figure 8).

STAGRAF Z too process one table in the set of sixteen, tracing separation polygonals between cells differently coded; user is allowed to choose interesting codes, to obtain partial zoning (Figure 9).

STAGRAF 16 works on the whole table set reading consecutively - if all existing an disk - 16 data files concerning the same variable; drawing on the same screen area - 160 for 160 pixels - program can only turn on pixels corresponding to cells qualified by user's selected codes (Figure 10).

All programs perform a superimposition routine, which allows matching of STAGRAF 1 drawing on a previously stored STAGRAF Z output and vice versa (Figure 11). As in COMP and CONF programs, a correspondence list of symbols and codes is automatically stored after each new table drawing.

Notes

[1] "Metodologia di analisi economico-territoriale perla individuazione delle suscettivita' produttive dei suoli - Ist. di Economia e Politica Agraria e Corso di Specializzazione in Pianificazione urbanistica applicata alle Aree Metropolitane dell'Università di Roma - La Sapienza"

[2] ADM design has been based on more complex packages (IMGRID, for instance), implemented on greater computers.

Acknowledgments:

Sergio Caldaretti has contributed to general package design;

Luca Martone has written earlier versions of CARSER, COMP and STAMP, then revised by author;

Franco Curatola has modified COMP an CONF programs, enabling multiple functions, and has written with the author RIAGR and INTOR programs.

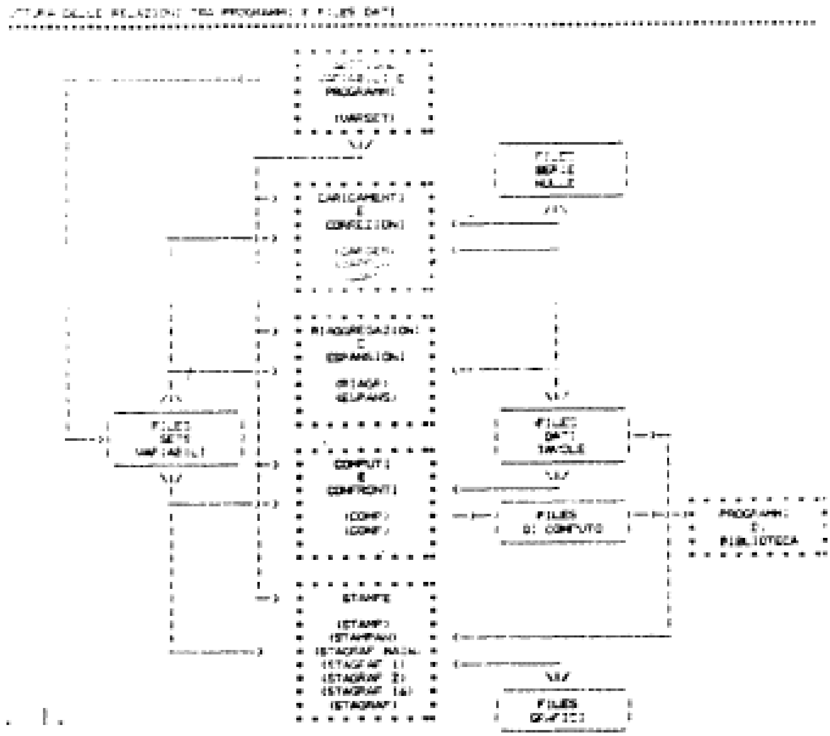
PROGRAMMI
(VARSET)
CARICAMENTI
E
CORREZIONI
RIAGGREGAZIONI
E
ESPANSIONI
FILES
SETS
VARIABILI
COMPUTI
E
CONFRONTI
(COMP)
STAMPE
(STAMP)
FILES
DATI
TAVOLE
FILES
DI COMPUTO
PROGRAMMI
DI
BIBLIOTECA
FILES
GRAFICI

FIG . 1.

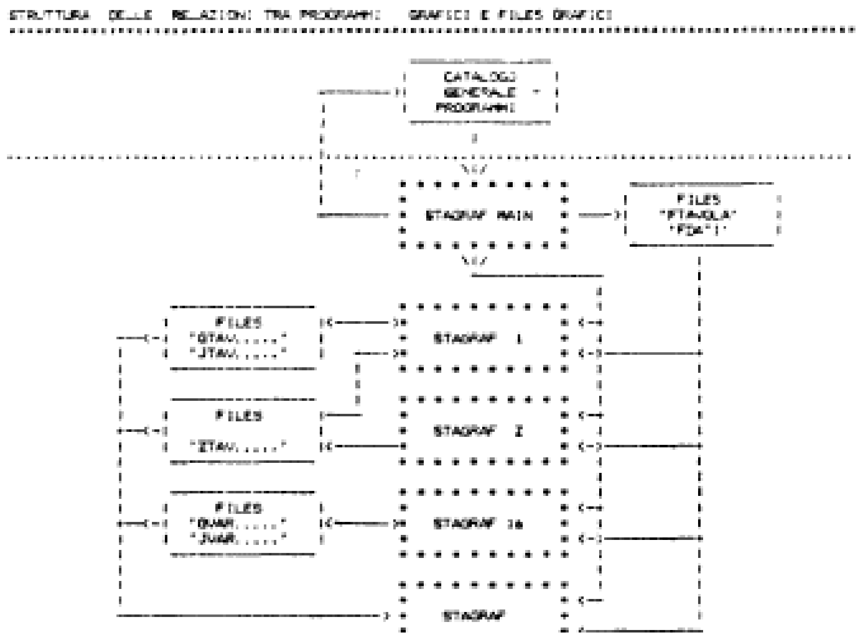
STRUTTURA DELLE RELAZIONI TRA PROGRAMMI GRAFICI E FILES GRAFICI
CATALOGO
GENERALE
PROGRAMMI
STAGRAF MAIN
FILES
STAGRAF 1
FILES
STAGRAF 2
FILES
STAGRAF 1a
STAGRAF

FIG. 2.

TAV.4 UAR.ALTIMETRIA/0

CL = CLASSI TAV.4 UAR.ALTIMETRIA/0 <73 CLASSI>
FR = FREQUENZA
NC = NUOVA CLASSIFICAZIONE
F1% = FREQUENZA % SENZA LA CLASSE 0
F2% = FREQUENZA % TOTALE

CL	FR	NC	F1%	F2%	CL	FR	NC	F1%	F2%
[illegible]	1	...	.06	.06	24	[illegible]	...	3.22	[illegible]
4[illegible]	1	...	.06	.06	[illegible]	[illegible]	...	[illegible]	3.1[illegible]
[illegible]	2	...	.12	.12	[illegible]	[illegible]	...	3.4[illegible]	[illegible]
[illegible]	4	...	.25	.24	[illegible]	[illegible]	...	3.54	[illegible]
[illegible]	4	...	.25	.24	8	57	...	3.67	3.56
36	6	...	.38	.37	26	57	...	3.67	3.56
31	6	...	.38	.37	28	62	...	3.99	3.87
[illegible]	7	...	.45	.43	19	65	...	4.19	4.06
34	11	...	.7	.68	20	69	...	4.45	4.31
32	12	...	.77	.75	16	72	...	4.64	4.5
27	19	...	1.22	1.18	12	77	...	4.96	4.81
[illegible]	20	...	1.28	1.25	14	83	...	5.35	5.18
30	21	...	1.35	1.31	17	97	...	6.25	6.06
25	26	...	1.67	1.62	15	110	...	7.09	6.87
18	34	...	2.19	2.12	11	118	...	7.61	7.37
23	40	...	2.58	2.5	13	265	...	17.09	16.56
21	45	...	2.9	2.81					

FIG. 3.

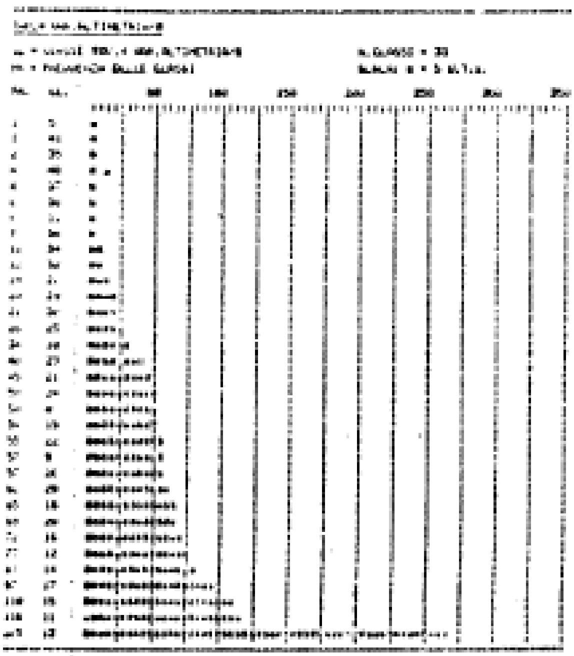

FIG. 4.

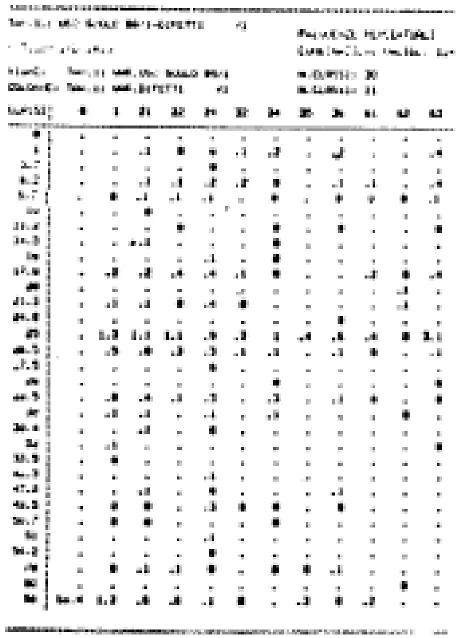

FIG. 5.

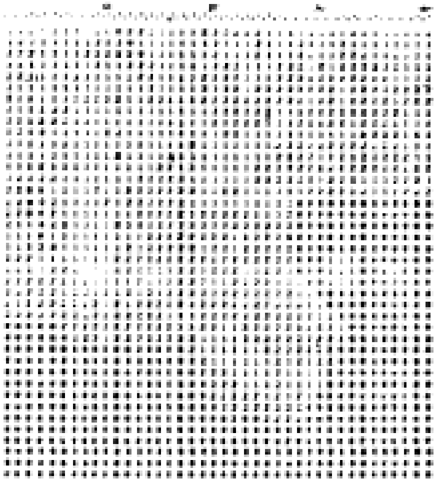

FIG. 6.

FIG. 7.

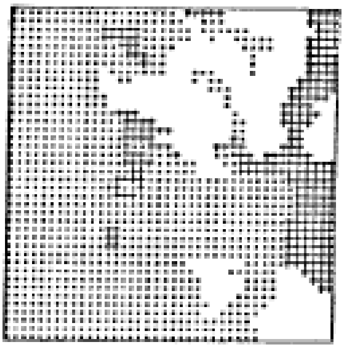

FIG. 8.

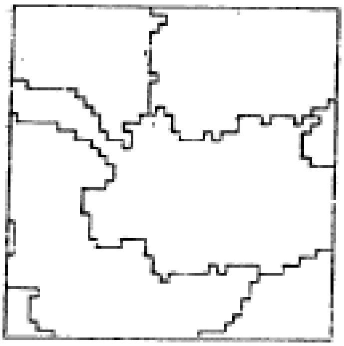

FIG. 9.

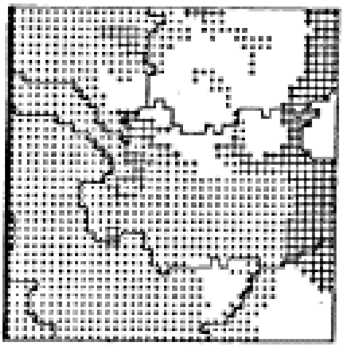

FIG. 11.

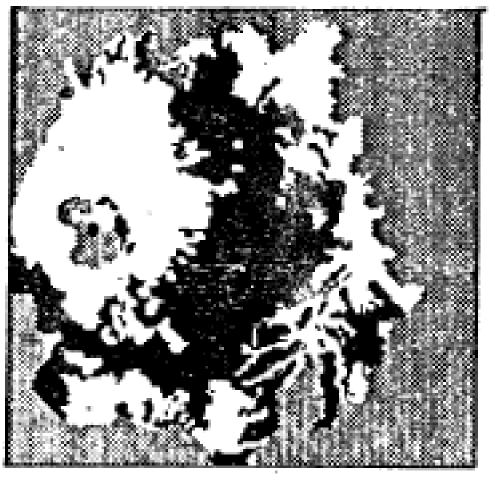

FIG. 10.