SPECIFYING THE CARTOGRAPHIC WORKSTATION HUMAN INTERFACE

Gordon K. Short
Systems Approach Ltd.

ABSTRACT

An important process performed at an automated cartographic workstation is the entry of instances of cartographic features. As in all other data capture and manipulation systems, both the format of the man-machine interactions and the format of the data specification can greatly affect the efficiency of the data entry process.

Due to the multitudinous variety of cartographic features, the entry of cartographic data is particularly prone to any adverse effects of an inefficient data entry process. The format of the data entry process in a cartographic workstation is therefore crucial to the success of the system.

This paper is concerned with the design and implementation of a tool that permits the end-user to specify an interaction process and a data format for the entry and modification of cartographic data. The benefits of using such a tool are explored from the point of view of both the applications programmer and the end-user.

DEFINITION DU LIEN ENTRE LE POSTE DE TRAVAIL CARTOGRAPHIQUE ET L'OPERATEUR

RÉSUMÉ

Un des procédés importants dans un système automatisé de cartographie est l'enregistrement des données cartographiques. Comme dans d'autres systèmes de prélèvement et de manipulation de données, l'organisation des interactions opérateur-machine et des spécifications peut grandement affecter l'efficacité des opérations d'enregistrement.

Vu la gamme variée de caractéristiques cartographiques, les procédés d'enregistrement de données sont particulièrement enclins à subir les effets de méthodes inefficaces. L'organisation des procédés d'enregistrement au poste de travail est donc cruciale dans la réussite du système.

Cet article décrit la création et l'implantation d'un outil qui permet à l'utilisateur de spécifier un mode d'interaction et une méthode d'organisation pour l'enregistrement et la manipulation de données cartographiques. Les avantages d'un tel outil sont examinés du point de vue de l'utilisateur et du programmeur.

SPECIFYING THE CARTOGRAPHIC WORKSTATION HUMAN INTERFACE

The traditional solution to providing an end user with appropriate process and data formats is to study the application and derive a set of operations and data formats that hopefully, will fully encompass the range of interactions and data types required. Finally, the design engineer may provide 'hooks' by which a suitably trained end-user may detail a new format of interaction or a new format of data specification that is to be included in the man-machine interface (just in case the design engineer missed an obscure point or perhaps the end-user's requirements will change over time).

Inexorably, the end-user's requirements will change, and the design engineer is bound to miss some obscure point in the end-user's requirments. Therefore, the end-user is destined to make some kind of modification to the man-machine interface.

The man-machine interface in an urban cartographic workstation is particularly prone to the need to evolve with changing end-user's requirements. The multitudinous variety of representation formats ensures that the interface will never be static. Indeed, the form of representation of a particular feature in an urban center may be so different from the form of representation in another urban center that even the starting definitions of the man-machine interaction repertory will vary greatly.

For the urban cartographic workstation, a dynamic man-machine interface is crucial to the success of the workstation. The interface must be easily tailored to each implementation center, and must be capable of easily evolving with the cartographer.

If it is given that the cartographic draftsman, as an end-user of an automated cartographic system, will need to change his tools for data entry, it is appropriate to ascertain the 'suitable training' required of the cartographer to perform the required changes. Following is a short list of the areas in which he must have a strong 'working knowledge'.

1) The end-user must know the computer language that the application program is implemented in. Few, if any of the leading cartographic systems allow the mixing of computer languages.
2) The end-user must know the internal data representation of the application program, as well as know of the internally defined utility routines for data manipulation.
3) The end-user must know how to interact with the operating system in order to compile and link the new routine.

The cartographer, as an end-user, does not care to bother with the above mentioned items. Clearly, a user-oriented method is required for specifying the interaction and data formats. The approach considered here is to provide an application oriented program for specifying the formats of both the interactions and the data that are to be handled by the man-machine interface.

The program that is supplied to the user for modifying the workstation's man-machine interface acts as a translator for the user. The translator must know the internal data formats of the system, the internally defined data manipulation routines, and the internally recognized man-machine interactions. The user may specify to the translator in a user-oriented syntax the formats of both the data and the interactions pertaining to a cartographic feature. The translator can then interpret the user's specifications to the man-machine interface.

A successful machanism for implementing the translator has been to segregrate the various data manipulation processes and man-machine interaction processes into named procedures. The translator can then translate the user's specifications into a series of procedure references. GENERATE is the software module in the Urban Mapping System which translates the cartographer's specifications for a feature template.

The translator approach is useful to the application programmer as well as to the cartographer. Firstly, the data and interaction formats may be comprehensively defined at the onset of the design project. Consideration of the design of the man-machine interface and the translator requires that the man-machine interaction vocabulary be rigorously defined.

Secondly, the translator can reduce coding errors by the application programmer during the creation of test systems. Normally for a test system the application programmer would be required to write several examples of feature templates. Each of the examples may have coding errors. By using the translator to write the feature templates, the application progammer is guarenteed a 'bug free' template, that is, he may expect the template to be a valid definition.

Another benefit to the application programmer is that this approach allows design changes in data entry programs or data storage formats which can be universally incorporated into the templates for all the features. This may be accomplished without too much difficulty by either re-running the translator with the same user specification or perhaps re-compiling the translator generated template.

GENERATE is the software module in the Urban Mapping System which translates the cartographer's specifications. The translation is a set of complementary functions which embody the formatting information for the specification, alteration, and storage of a cartographic feature instance. The complementary functions are automatically compiled and

link-edited into the Integrated Cartographic Editor. To paraphrase, GENERATE allows the cartographer to create a template that will be incorporated into the Integrated Cartographic Editor. That template embodies the procedure and format for the definition of each instance of that particular cartographic feature.

The first step towards the definition of the man-machine interface and the GENERATE program is to ascertain the data storage formats and the interaction processes that pertain to a cartographic workstation. These processes and data storage formats can then be embodied as primitives in the man-machine interface GENERATEd template.

Two types of data, graphical data and non-graphical canned attribute data, are used in the Urban Mappin System to describe cartographic features. These two types of data are differentiated between because canned attribute data can be defined in a predicatable series of interactions and stored in a fixed number of fixed-length fields, whereas graphical data may consist of an unknown order of unknown number of variable-length fields. For example, the non-graphical attributes of a house may be described as an ordered set of fields such as tax appraisal value, address, and construction material, but can the graphical attributes of a house always be a fixed number and order of geometric figures?

Three types of graphical data and four types of attribute data are recognized in the Urban Mapping System. The three types of graphic data are AREAL, LINEAL, and POINT. The four types of non-graphical attribute data are INTEGER, FLOATING, TEXT, and SELECT. The man-machine interface has a named procedure for the defining and editing of each of the these seven types of data.

The cartographer uses the program GENERATE to create the feature template that utilizes the named procedures in the man-machine interface. GENERATE is a menu driven program operated from a graphical input/output screen and a keyboard. Upon initiation, the GENERATE program will query the cartographer for a feature name. This name will be included in the man-machine interface as an identifier for the feature template.

Once the feature name is known, the detailing of the feature's data domain and format may begin. GENERATE will present the user with the following menu:

SNAPGRID	snap the grid to the baseline of the instance
SETGRID	set a default grid for each instance
GETAREAL	get areal graphics
GETLINEAL	get lineal graphics
PUTSYMBOL	put the feature symbol at the indicated point
MESSAGE	print a message to the cartographer
ATTRIBUTE	get canned attributes
EXIT	terminate specification of the template

The digitizing of any menu item will cause the associated data types and/or actions to be included with the feature template. Each of the

menu functions may poll for additional information and include it in the feature template. The graphical function will request the overlay layer and the pen type that the graphical information is to be associated with. The MESSAGE function will query the user for the message to be displayed to the draftsman during the feature instance definition.

The attribute function will query the user for the number of data fields to reserve. GENERATE will then cycle through the fields in sequence so that each may be defined. A menu of data types will be displayed for each field consisting of:

TEXT - the width of the field will be polled for

INTEGER - the minimum and maximum allowed entries will be polled for (negative and positive infinity are default)

FLOATING - the minimum and maximum allowed entries will be polled for (negative and positive infinity are default)

SELECT - keyword values are used to specify the contents of the field. The keywords may also have alternate forms such as abbreviations. Each select value will be assigned a value in ascending order and only the counter will actually be stored in the attribute record. The first keyword entered in the synonyms list will be the displayed word on the map if the field is so included.

As each field is defined the cartographer may also indicate that the value of the field will be placed on the map. GENERATE will cycle through all such flagged fields allowing the cartographer to graphically specify the display location of the field value (relative to the origin of the feature instance), the angle of the display value (relative to the snapped grid if there is one, relative to the horizontal otherwise), and an output format type.

To specify the format of the attribute display information, the cartographer will be asked for the dimension of a square that will encompass a typical instance of the feature. That square will be shown full screen as a reference for the cartographer to specify the text location. The menu will have the following items:

GRID - to change the grid size in digitizing the text locations

LAYER - to specify the layer of the text

PEN - to specify the pen of the text

TEXT - to specify constant text to be placed on the map

APPEND - to specify constant text that is appended to a field that is displayed on the map

FORMAT - to specify the output format of the value of the field that is diaplayed on the map

As each field is defined, a rectangle outlining the text will appear on

the screen. Instructions to the cartographer detailing the format of the field will appear on the screen.

Not enough experience has yet been gained in the use of the translator GENERATE. Although considerable success has been realized with the generalized feature template approach to detailing the interaction and data formats for the specification of instances of cartographic features, the implementation and interaction formats of the GENERATE program have yet to be put on trial.