First UNIX, then UGIX

D.W. Rhind, J.F. Raper and N.P.A. Green

Department of Geography, Birkbeck College,
University of London,
7-15 Gresse Street, London W1P 1PA, UK

ABSTRACT

This paper proclaims the need for increased standardisation within the field of GIS and defines some assumptions about the way in which GIS systems will evolve and develop if insufficient attention is paid to standardisation. Building upon these assumptions, we describe how the existing GIS Tutor can be expanded into forming a vital part of a Universal Geographic Information eXecutive (or UGIX). The initial role of this is to act as a 'friendly front end' to any existing GIS and to free the user from having to learn new conventions, rules and grammar every time s/he works on a different system. Subsequent, higher order aims are also described.

STANDARDISATION, USERS AND VENDORS

We begin with a conjecture : the simpler and more standard are computer systems, the more readily they are used and the more choice of system is open to the user. Standardisation, at least of basic tools and methods of using them, is therefore 'a good thing', so far as the user is concerned.

A major theme of the last 10 years in the data processing industry as a whole has been that of standardisation; Nash and Redwine (1987) have identified over 1000 software-related standards in the USA alone. Perhaps the most striking example thus far has been the popularisation and gradual acceptance of the UNIX operating system and, latterly, the requirement in many defence contracts that the systems are POSIX - compliant. In the medium to longer term, Open Systems Interconnect (OSI) might be even more important. Though some GIS now capitalise upon UNIX, some provide querying through the Structured Query Language (SQL) and some systems use the Graphics Kernal System (GKS) or the Programmers' Hierachical Interactive Graphics System (PHIGS) as graphic output mechanisms, no significant standardisation has gone on within the GIS field except in terms of data description and transfer protocols (e.g. ACA 1988 and OS 1987).

It is worth considering why this should be so and, in so doing, noting the present dominance of commercial concerns in the field. This is a recent phenomenon : the primary difference between Rhind's 1981 and 1987 reviews of the GIS field in the UK, for instance, was the almost total lack of any commercial presence observable at the earlier date yet the domination of the field by such interests at the latter. What we have seen, therefore, is an emerging market place characterised by increasingly strong competition amongst an increasing number of vendors. No one of these vendors presently seems prepared (or, in some cases, is financially able) to move to agreed standards - even if the latter existed.

In reality, we should not be surprised at a lack of standardisation : the user base is manifestly so diverse, rapidly expanding and disorganised that it has been difficult to define a set of procedures, addressing mechanisms and conventions which cover user needs, let alone exert pressure on the vendors (except through industry - wide groups, of which the National Joint Utilities Group in the UK is a striking example). As a consequence, the vendors have little incentive to offer standardisation. Indeed, taking a longer term view, the users can be considered to be enjoying a period of warfare amongst vendors which will, through the operation of the market, result in a few successful suppliers; the products from the survivors could well form the de facto standards in years to come. Adopting this viewpoint, we - as users - should simply sit back and wait for Adam Smith's "invisible hand" to ensure the appearance of a set of standards.

We eschew this passive approach : we hold that the user has a right, even an obligation, to assist and encourage vendors in moving towards standardisation - provided that this does not stultify new developments. Without such efforts, a small number of vendors may come to dominate the market with different proprietary solutions and to 'lock in' the user to their particular products. Precisely this occured in the 1960s and 1970s in data processing as a whole. We therefore advocate an activist and even interventionist approach, stressing the primacy of the user rather than the supplier. We make several assumptions which seem reasonable and proceed from these to recommending a course of action.

The first of our assumptions is that many GIS users will increasingly regard their data bases as a long term and possibly appreciating asset; in contrast, they will treat their GIS software as an asset which is depreciated normally. At present, this distinction is difficult since data are often intimately wrapped up in proprietary features in any one GIS. The second of our assumptions is that competitive pressures will force convergance between the solutions being offered by vendors, at least at the levels of data structure and functionality; Rhind and Green (1988) have summarised the advantages of different data structures and GISWorld (1988) has published a table of functionality claimed by vendors which seems to demonstrate the progression to functional equivance as being underway.

Given all of the above, we anticipate that users - in GIS as elsewhere - will seek freedom to purchase the best deal available as they purchase second and third generation GISs (perhaps to meet new tasks or as software suppliers go out of business) and as costs and capabilities change. At present, this is rendered impossible by the different functionality, different data structures and the often idiosyncratic and painfully acquired knowledge of how to 'drive' any one system. To facilitate the change requires at least (as we argue later) an 'intelligent front end' which can speak to all GISs in their own command languages yet which can be instructed by the user in a universally accepted 'geospeak'.

Before we set out the components of a GIS which are amenable to standardisation, however, we describe how we see the market developing; this has important implications for the starting point and capabilities of the UGIX which we describe later in the paper. To do this, we extrapolate from the example of a more mature and widely used product than any present-day GIS.

dBASE AS AN EXAMPLAR FOR THE GIS MARKET

The most dramatic change in computing in the last 10 years has been the growth of local computing power in comparison with that in centralised machines. Thus the availability of cheap yet powerful micros fueled and, in turn, benefited from the availability of general purpose software packages. Some indications that the GIS market is already going much 'more personal' already exist : the dramatic success of ESRI, for instance, in selling 1300 copies of PC ARC/INFO in its first year, compared to about 500 copies of mini- and mainframe versions over four years, when allied to the success of SPANS and other micro- based and workstation- resident systems, suggests that GIS is merely following the trend set by systems such as Wordstar, Lotus 1-2-3, dBase and many others.

Though the analogy of its evolution with that of GIS systems is not exact, the story of dBASE is directly relevant to our concerns, not least because it has now sold over 100,000 copies and has become nearly ubiquitous. It was 'invented' by Wayne Ratcliff, being developed in his spare time to keep track of football results for the office sweepstake system. The result was entitled Vulcan and, when marketed, sold very badly. Only when Ashton-Tate took over the marketing did matters change; they initially sold it as dBASE II for use on CP/M machines. Later, Ashton-Tate bought the rights to dBASE though Ratcliff stayed on as Vice-President and in charge of development for version III. Following disputes about the way in which dBASE should develop, Ratcliff left the company. dBASE IV has recently been launched.

The nature of the product has evolved dramatically over the years. Version II was essentially a programmer's toolbox. It consisted of just over 100 commands, each of which was activated via use of the dot prompt. Restrictions were numerous e.g. a limit of 65,535 records, each of which could have up to 32 fields; a very limited form of relating together two files was provided. Version III, in contrast, provided a menu- driven package called the Assistant; it expanded the number of commands by about 35%, introduced set relational capabilities and relaxed many of the more irksome restrictions and facilitated the creation of user-designed screens. The extension, dBASE III Plus, introduced more powerful commands for programmers and improved the Assistant. Finally (at least thus far), dBASE IV provides multiple user interfaces, an increase in speed and improved networking capabilities. In particular, the Assistant has been replaced by the Control Centre which enables those users who so choose to use dBASE as an entirely menu-driven system. It provides Query By Example capabilities and an applications generator.

An SQL (Structured Query Language) interface has been added whilst the original dBASE idiosyncratic commands may still be used.

Several relevant conclusions may be drawn from the dBASE story. These include :

- that early systems which become successful encourage both lower cost clones and 'add-ons'; through this, they may become de-facto standards

- though many successful systems start out designed for experts, they end up catering for a mass market

- the size of evolving packages normally gets larger and larger as the price both of ensuring upward compatibility and of providing new features. An important (and expensive) part of the additional features is likely to be an accomodation of standards initially ignored or recently promulgated.

- fortunately, the recent annual growth in computing power per unit cost has exceeded the rate of growth of size of software systems and also the size of 'average' applications (note that the latter statement does not apply to the largest GIS applications, leading to a divergence in the computing needs of 'average' and large scale GIS users (e.g. global modellers)). dBASE succeeded by enabling average size applications to be run - and continue to be run as these applications became larger - on contemporary micro- computers.

- in dealing with a mass market product, superb documentation, highly robust software, training materials and a secondary 'value added' industry are essential to provide success

- in many respects, the adoption and spread of data base systems developed along similar, but earlier, lines to those of GISs. Given this precedent and the ready availability of mass market graphical, data base and other software, it seems reasonable to expect PC GISs soon to be selling for little more than the combined cost of dBASE and, say, the Harvard Graphics package (c. #750 at list price in the UK). If standard data bases are also available, the consequences of such pricing for GIS sales will be immense.

GIS ELEMENTS AMENABLE TO STANDARDISATION

These would seem to be :

(i) the terminology used to describe individual functions, data elements, etc

(ii) the form and structure of data dictionaries

(iii) a set of standard tasks for testing GIS, with mathematically provable end - results

(iv) a library of proved routines, using the best available algorithms. This would be the equivalent of the Numerical Algorithms Group (or NAG) Library. An early attempt to bring this about was the Geographical Algorithms Library or GAG (Campbell 1977)

(v) standard human interfaces to GIS; as Rhind and Green (1986) and others have pointed out and as dBASE, Domesday (Rhind et al 1988) and other systems have implemented, multiple access routeways are needed for different users and types of applications. These will inevitably include WIMPs (with standard ikons), an SQL interface and simple sets of menus. More important, it should include a 'Geospeak' language since SQL is ill-suited for spatial queries yet many individuals prefer to express their requests in a written rather than a graphic language. Egonhofer and Frank (1988), Palmer and Frank (1988) and Goh (forthcoming) have begun the process of designing spatial query languages.

(vi) data transfer formats and protocols. Much work has already been carried out in this area.

From all of this (and other evidence), we conclude that mass market GISs are likely to form THE major growth area and that this will be based upon Mac, PS/2 or DEC architectures. This is manifestly a technical possibility : existing packages with the power of 386 processors, allied to Winchester - type storage of 100 to 300 Mb and archival storage of 600Mb per exchangeable CD-ROM, give capabilities available a few years ago to only the most priviliged. We anticipate, therefore, a growing number of PC and workstation - based GIS over the next few years. Thus we believe that action is necessary now and that the best way to bring about standardisation should be :

(i) international action, involving multi-disciplinary inputs, to tackle items (i) to (iv) and (vi) in the list of elements amenable to standardisation.

(ii) a concentration on 'binding in' UGIX to the micro-based systems (although it must also be available as a micro-based 'front end' which can converse with and issue instructions to mini- and mainframe- based GIS). This therefore addresses the final element in our list and the rest of this paper is devoted to this topic.

THE uNIVERSAL gEOGRAPHIC iNFORMATION ExECUTIVE (OR UGIX)

We see UGIX as an 'intelligent front end' which can speak to all GIS in their own command languages yet which can be instructed by the user in a universally accepted and standard 'Geospeak'. A necessary preliminary to discussing UGIX, however, is to define a conceptual framework for the range of options available in GISs. This can be obtained from various different sources; table 1 summarises the findings of Rhind and Green (1988), and shows a superset of known functions, albeit described in general terms. GISWorld (1988) has provided a mapping from a similar list of basic functions to existing GIS (or, at least, what the vendors claim their systems can achieve).

Essentially, the design for UGIX - the entire section within the main box - consists of three sub-systems, labelled A, B and C in figure 1. UGIX provides, in its most basic form, a command-level interface to (in theory) a variety of extant systems. In practice, we intend to build the prototype with only two system interfaces in the first instance. The contents of the boxes are :

Box A

The Decision Taker is the heart of UGIX. Upon receipt of commands or of output, it decides what resources and other information are needed to effect its tasks. Within it, are four sub-modules.

System Configuration module. This stores details of the UGIX configuration itself and of the systems with which it can communicate.

The Dialogue Handler. This handles all interaction between user and all sub-systems; thus it handles all input commands and also all reports (including graphic portrayals) resulting from these commands. It provides multiple methods of issuing commands but provides a 'point and click' default.

The Rules Table. This contains all the rules and protocols which control the actions of the Decision Taker.

The Mapper. This converts all the UGIX commands into those required to drive the other systems (where this is possible given their capabilities) and converts all output from them back into UGIX reports.

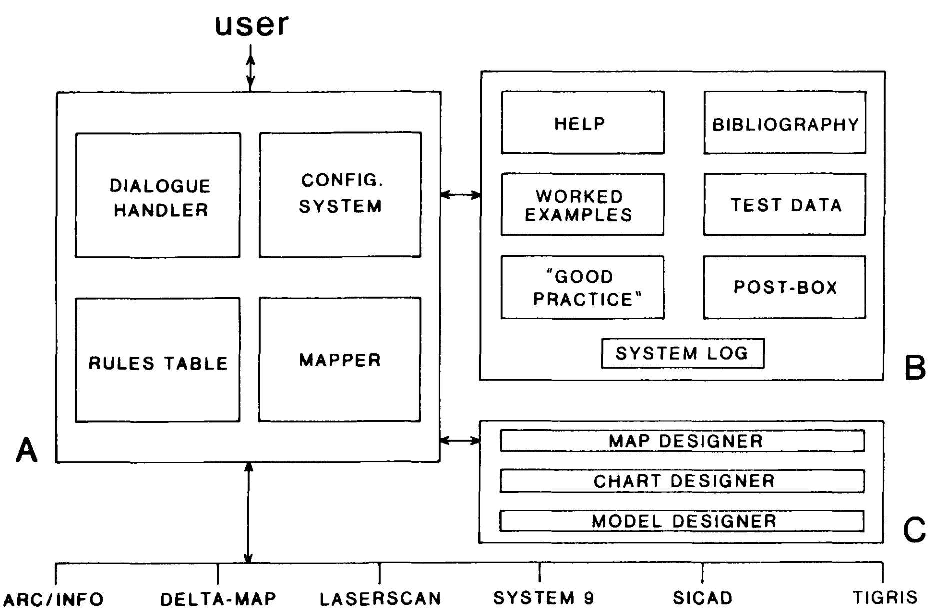
Box B

This consists of an enhanced version of GIST (Raper and Green 1989). In it is a sub-module for providing Help facilities (in principle in regard to all the systems with which it UGIX is linked, as well as the host system itself and GIS concepts, a glossary of terms, etc). Other sub-modules provide worked examples, a description of 'good practice' in carrying out the types of analysis being requested, some test data sets, a bibliography searchable on key worded topics, a log of all actions carried out by the system and the results and a post box for leaving messages (over the network if this is set up) for the postmaster on problems encountered. The messages may be initiated either by the user or by UGIX.

Box C

This module is the lowest of our priorities at present. It is intended to add 'intelligence' to UGIX's capabilities. We envisage at least three sub-modules, each with a suffix of 'designer'. Thus one sub-module provides not only within UGIX definition of scaling, clipping, feature selection and symbolism but also more 'intelligent' input such as context-setting (Egenhofer and Frank 1988); its capabilities approximate to those of the 'ideal mapping package' (Blatchford and Rhind 1989). Another sub-module provides a similar (though simpler) function for design of histograms, graphs, scatter plots, etc and the third is an interactive model design facility which permits the specification and calibration of user-specified models.

Figure 1 : functioning parts of UGIX



Many of these facilities already exist in one form or another. Thus far, we know of no situation where they have been brought together in a form approximating to UGIX. Our initial intention is to regard UGIX only as a language translator between 'UGIX-speak' and the specific vocabularies, grammars and other conventions used by different systems. In the longer term, we have greater ambitions for UGIX : we would wish to provide an option whereby users could instruct it in any of the languages used by any of the 'recognised' systems and drive any one of the others. This presupposes, of course, that the concepts in each system are mappable to each other system. Where this is not the case, the Rules Table and the Mapper will ensure that no attempt is made carry out the impossible. Finally, we would wish to use UGIX as a data translation tool for those circumstances where data had to be translated from one system's representation into another.

CONCLUSIONS

UGIX exists only in parts at present. To make it work as outlined will require resources in excess of those currently available to us if it is to be made available over a time scale of a year or two. We believe - and have good evidence to support the belief - that it could be built most speedily using Hypercard and other facilities for MacIntosh computers, especially if allied to tools such as SequeLink. We do not, of course, underestimate the task : it will, for instance, involve keeping up - to - date with new developments in different GIS systems. For this reason, we would be happy to embark upon UGIX as a collaborative venture, especially with those individuals or groups who have experience of systems of which we have no expert knowledge.

Table 1. A classification of GIS functions needed for UGIX

Data Input and Encoding

Data capture (eg. manual or automatic digitising)

Data validation and editing (eg. quality checking)

Data storage and structuring (eg. construction of link/node topology, chain coding, etc.)

Data Manipulation

Structure conversion (eg. vector-to-raster, quadtrees to vector)

Geometric conversion (algebraic and 'rubbersheet')

Generalisation and classification

Enhancement (eg. edge enhancement, line fractalisation)

Abstraction (eg.calculation of centroids, Thiessen polyons)

Data Retrieval

Selective retrieval of information based on spatial or thematic criteria, including 'browse' facilities.

Data Analysis

Spatial analysis (eg. polygon overlay, route allocation, intervisibility, slope and aspect calculation)

Statistical analysis (eg. frequency analysis, dispersion)

Measurement (of lines, areas, volumes, distance, direction)

Data Display

Graphical display of maps, graphs, etc. on both graphical display and on hard copy devices.

Report writing and progress messaging

Database Management

Integrated database management facilities include: support and monitoring of multi-user access to the databases; provision of 'roll-back' facilities for use in the event of system failure; oganisation of the database for efficient storage and retrieval without data redundancy; automatic maintenance of database security and integrity; providing the user with a 'data-independant' view of the database.

Note: This classification is derived from that of Rhind and Green (1988) and that in turn was based partly on the work of various other authors (see their paper for details)

REFERENCES

ACA (1988) The Proposed Standard for Digital Cartographic Data. American Cartographer, 15, 1, 137pp.

Blatchford R. and Rhind D.W. (1989) The ideal mapping package. In Rhind D.W. and Taylor D.R.F. (eds) Cartography - yesterday, today and tomorrow, Elsevier, Amsterdam

Campbell W.J. (1977) Computer algorithms for spatial data. Area, 9-2, 106-8.

Egenhofer M.J. and Frank A.U. (1988) Designing object-oriented query languages for GIS : human interface aspects. Proc. Third Intnl. Sympos. on Spatial Data Handling, 79-96, Intnl. Geographical Union, Sydney.

GISWorld (1988) GIS Software Survey. GIS World, July, 7-11

Goh, Pong-Chai (forthcoming) A graphic query language for cartographic and land information systems. Intnl. J. of GIS

Nash S.H. and Redwine S.T. (1987) A map of the world of software-related standards, guidelines and related practices. Computer Standards and Interfaces, 6.

OS (1987) The National Transfer Format, Ordnance Survey, Southampton, UK.

Palmer B.L. and Frank A.U. (1988) Spatial languages. Proc. Third Intnl. Sympos. on Spatial Data Handling, 201-9, Intnl. Geographical Union, Sydney

Raper J.F. and Green N.P.A. (1989) GIST : an object-oriented approach to a GIS Tutor. Proc. Auto Carto 9

Rhind D.W. (1981) Geographic Information Systems in Britain. In N.Wrigley and R.J.Bennett Quantitative Geography, 17-35, Routledge and Kegan Paul, London.

Rhind D.W. (1987) Recent developments in GIS in Britain. Intnl. J. of GIS, 1, 3, 229-41.

Rhind D.W., Armstrong P.A. and Openshaw S. (1988) The Domesday machine : a nationwide GIS, Geogr. Jl., 154, 1, 56-68.

Rhind D.W. and Green N.P.A. (1988) Design of a GIS for a heterogeneous scientific community. Intnl. J. of GIS, 2, 2, 171-89.