Available online at www.sciencedirect.com

ScienceDirect

Procedia Social and Behavioral Sciences 21 (2011) 3–9

Procedia
Social and Behavioral Sciences

International Conference: Spatial Thinking and Geographic Information Sciences 2011

Spatial Thinking and the GIS User Interface

Michael F. Goodchild[a*]

[a]Department of Geography and Center for Spatial Studies, University of California, Santa Barbara, CA 93106-4060, USA

Abstract

Geographic information science can be defined as the study of the fundamental issues of geographic information, and is often motivated by the need to improve geographic information technologies. One such issue concerns the design of the user interface, and the relationship between the tasks performed by the technologies on the one hand, and the concepts that humans use in thinking about those tasks on the other. Nowhere is this issue more important than in the design of GIS user interfaces and functionality. Recent efforts have led to a comprehensive understanding of the concepts of spatial thinking, and of how these concepts might form the basis for a much-improved functionality and user interface. The presentation summarizes those efforts, and points to a future in which GIS will be much easier to teach, master, and use.

Selection and/or peer-review under responsibility of Yasushi Asami

Keywords: Spatial thinking; Geographic information system; User interface; Spatial concepts;

1. The Problem

Geographic information systems date from the 1960s, when the first attempts were made to apply computing systems to the handling and processing of geographic information. The Canada Geographic Information System (CGIS), for example, was developed in that era with the objective of capturing and storing the contents of a large number of maps, and producing statistics on the amounts of area mapped as having specific characteristics. For example, a map of land use would be analyzed to determine the areas used for agricultural purposes. The system had very limited functionality: in addition to basic housekeeping, the system supported queries about the characteristics of specified points, and summaries of areas within specified shapes such as circles centered on specified points.

By the late 1970s various initiatives to apply computing to geographic information had converged, into a vision for a system that could perform a wide range of functions on many different data types. Applications in automated cartography, the management of the census, and land-use planning had converged with that represented by CGIS; and geographic information types included representations of road networks and topography. An industry dedicated to the creation and marketing of GIS software had emerged in the early 1980s, and since then GIS has continued to flourish. Today it is fair to say that the

* Corresponding author. Tel.: +1-805-893-8049; fax: +1-805-893-2578.
E-mail address: good@geog.ucsb.edu.

1877–0428 © 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of Yasushi Asami
Open access under CC BY-NC-ND license. doi:10.1016/j.sbspro.2011.07.002

leading GIS software products are capable of virtually any conceivable operation on any recognized type of geographic information. Functions have been added to address the needs of specific disciplines and problem areas, to integrate data that has been prepared using a wide range of standards, and to support a large number of forms of analysis, modeling, and visualization.

Nevertheless, this incremental process of development has created a number of problems. GIS has developed without a well-defined and established body of theory; by contrast, the development of statistical software has always been able to rely on formally defined terms, and on techniques that were developed long before the advent of computers [1]. Many GIS operations are defined by reference to the physical maps that represented the primary storehouse of geographic information prior to computerization. Thus the integration of GIS data sets is commonly termed "overlay", because of the physical analogy of the overlay of transparent maps, each representing a "layer" of information, though I comment in a subsequent section about whether this GIS task actually requires invocation by the user. The map metaphor persists in GIS, and it is still difficult to represent and work with geographic information that is not suited to the traditional map: dynamic data, for example, or three-dimensional data, or data about the interactions of *pairs* of places that I term *binary* geographic information to distinguish it from the more map-like *unary* information, or information about locations taken one at a time. The binary category includes flows, migrations, travel times, and distances. Other operations mirror those of computer science, including the "join" that is well recognized in relational algebra. Still others are defined by their objectives and outcomes, rather than by the conceptual intent of the operation.

To take one example, the current version of Esri's ArcGIS desktop software ArcMap 10.0 includes hundreds of operations. Some are invoked through drop-down menus, while a much larger number are invoked through the Toolbox. The total number of operations in the Toolbox is 615. Organizing them into a logical structure has proven very difficult. The current Toolbox organizes them hierarchically, into 18 top-level headings that include "3D Analyst Tools", "Analysis Tools", "Geostatistical Analyst Tools", "Spatial Analyst Tools", and "Statistical Analyst Tools". These top-level headings overlap substantially, making it very difficult for a user to decide which heading to explore in a given instance, or to learn the software in an educational setting. The number of tools under each of the top-level headings ranges from 2 to 178; this fact alone suggests that the structure could be improved.

In short, the forty-year process of development of GIS has led to a functionality that is confusing, poorly defined, and lacking in any conceptual framework or organizational structure. The purpose of this paper is to explore some of the ways in which such a framework or structure might be developed. If successful, it would lead to a substantial simplification of the user's interaction with GIS, a substantial simplification of the task of learning GIS skills, and a substantial contribution to the GIScience literature.

In addition to ease of use and learning, there is a second motivation for addressing the complex functionality of today's GIS. Service-oriented architecture (SOA) is emerging as a new paradigm of computing, in which tasks are performed by chaining together on-line services rather than by operations at the desktop. In order for SOA to work it is essential that it be possible to search for and discover services, to evaluate them for suitability, and to invoke them. This process of search presumes the existence of standard terminology, so that the user can identify precisely those services that meet his or her need. But at this time there is no such standard terminology, despite earlier efforts. The functions defined by each GIS reflect numerous factors, including the legacy of previous versions of each software package, and other factors discussed earlier. Thus a new approach to functionality, that formalizes it in order to create standards, is essential if the promise of SOA is to be fulfilled in the GIS domain.

Various efforts have been made by the GIS research community to bring some semblance of order to this apparent chaos. Of these, perhaps the most successful is the *map algebra* developed initally by Tomlin [2] and similar in many respects to the *image algebras* of image processing (*e.g.*, [3]). Operations on raster representations were organized into four distinct types: *local* operations performed on each raster cell or pixel independently; *focal* operations which compared each pixel with its immediate neighbors; *zonal* operations performed on contiguous cells sharing a common attribute; and *global* operations on all cells. The scheme was sufficiently general to be adopted by several software packages, including elements of Esri's ArcGIS. Significant extensions were later made by van Deursen [4], Takeyama and Couclelis [5], and others. Note, however, that the scheme still places emphasis on the algorithmic operations of each function, rather than on its conceptual goals.

Several taxonomies have also been proposed (see, for example, [6], [7], [8], [9]), though none is substantially more satisfactory than the ArcGIS taxonomy discussed earlier. Goodchild [10] grounded his taxonomy in the GIS's database, arguing that the latter formalized the representation of spatial data, and thus could provide a more formal grounding for a taxonomy of functions. His taxonomy is limited to discrete-object representations (for a discussion of the field/object dichotomy see, for example, [11]). In Unified Modeling Language (e.g., [12]):

- Analyze the attributes of a single class of objects (statistical analysis);
- Analyze one class of objects using both locational and attribute information;
- Analyze the attributes of an association class;
- Analyze more than one class of objects;
- Create a new association class from existing classes; and
- Create a new class from one or more existing classes.

The results of further analysis along these lines were presented in a later paper by Burrough [13]. With further development, one might seek to devise a test of the completeness of this scheme, and to create a more formal description.

The next section describes three ways in which the interface might be simplified: by implicit invocation, by removing the need to interact with the low-level components of GIS representation; and regarding analysis as a set of transformations of a geodatabase. This is followed by a section reviewing recent work in enumerating the concepts of spatial thinking, and how they might form a more coherent basis for user-interface design and the organization of functionality. The paper ends with a brief conclusion section.

2. Three Strategies for Simplification

2.1. Redundancy

Consider the task known as polygon overlay, a primary feature of CGIS and of every GIS since. In the vector version, two area-class maps defined by non-overlapping, space-exhausting polygons must be combined to produce a single area-class map. Call the two input maps *red* and *blue*, and the output map *purple*. Then the purple map contains all of the boundaries of both the red and blue maps, except where red and blue boundaries coincide. Since coincidence will never be perfect in independently produced maps, some procedure must be implemented for removal of *spurious* polygons and the merger of their boundaries. Attributes of the red and blue polygons will be concatenated to provide the attributes of the purple map. The process is reversible, though removal of spurious polygons will produce variations in the locations of some boundaries after reversal. Note that polygon overlay is much more than simply drawing the two maps on top of each other and allowing the user's eye and brain to combine them, a operation variously known as a form of mashup or more traditionally visual overlay.

The programming of polygon overlay presented a major challenge to early developers. It is computationally intensive, though it responds well to various indexing strategies. A robust code for the task had become available by the early 1980s, and had much to do with the initial success of Esri's ARC/INFO. The task has become iconic in GIS education, and would be cited by many as one of the most central tasks of GIS. Polygon overlay can be presented as a special case of *spatial join*, the integration of two data sets that differs from a conventional relational join because the common key needed to integrate their objects is not available; instead, it must be inferred from spatial relations such as containment or overlap. Like a relational join, a spatial join is a form of de-normalization of a relational design.

It is surprising, therefore, to discover that user invocation of polygon overlay, or spatial joins in general, is never actually required, since the need for it is always obvious from a previously defined user query. In GIS education, students are often challenged to see that a given conceptual query requires the invocation of this specific GIS function; if the invocation could be automatic, this challenge would disappear. Consider, for example, the major function performed by CGIS, and the function that motivated its development: the calculation of the area having characteristics such as *capability for agriculture class 1* and *current land use class B*. To obtain such a statistic manually it would be necessary first for one map in transparent form to be overlaid on the other, and then for some method of manual area measurement to be used, such as dot counting or the use of a mechanical planimeter, both methods being tedious, labor-intensive, and error-prone. Instead the operation is fast, cheap, and accurate in a GIS, once the two layers have been topologically overlaid.

The spatial join or topological overlay is necessary in this case only if the two attributes, capability for agriculture class and current land use class, are not already available for the same set of geographic features—in the terminology of geostatistics, they are not available on a common *support*. Because they are not, since areas of uniform capability for agriculture are not normally mapped as congruent with areas of uniform land use, it is necessary to perform a topological overlay and to obtain a set of purple polygons that are homogeneous on both properties—in other words, to create a set of features that have both characteristics (see [11], Chapter 14). The operation of area measurement is then straightforward.

In response to the query "How much land is of capability for agriculture class 1 and current land use class B?", the GIS needs first to ask whether both attributes are available in the same attribute table, in other words for the same set of features. If they are not, then a topological overlay is clearly necessary. But this question can be resolved automatically given a standard database design that tracks the names of attributes. If the answer is no, the GIS can then invoke the spatial join automatically. Explicit invocation of the spatial join is never necessary in a GIS with an intelligently designed user interface. One is tempted to ask whether other GIS operations are similarly redundant at the level of the user interface.

2.2. Operations on Fields

GIS uses six common ways of representing continuous fields: as a collection of randomly located sample points, as a triangular mesh, as an array of rectangular cells, as a rectangular grid of regularly spaced sample points, as a collections of non-overlapping, space-exhausting polygons (the area-class maps of the previous section), or as a collection of digitized isolines. In querying or analyzing the field, one would not want to have to interact with these individual features, since their role is merely to support the representation. There is one exception, however, when the boundaries of the area-class map are established through some quite independent process, as is the case when the area-class map records values for a set of independently established reporting zones, such as counties or census tracts. Smith and Varzi [14] have termed these *fiat* boundaries in contrast to the *bona fide* boundaries of maps of land use or capability for agriculture.

Kemp [15], [16] has shown how entire fields can be manipulated and analyzed with single commands, extending the map algebra of Tomlin [2] that is specific to the raster case (rectangular cells or a rectangular grid). The user is freed from interacting with the individual features, and also with which of the six forms of representation is used. In this concept, a field A might be combined with a field B to obtain a field C by issuing the command C = *f*(A,B) where *f* represents some mathematical function, such as addition. The command would not need to specify the specific form of representation used by either A or B, and the system would be capable of making intelligent decisions about the form of representation of the output field C. If such decisions require user input, the system would automatically request a response. Again, this approach has the advantage of greatly simplifying the complexity of the user interface by removing the distinction between many different commands, and requiring the user to work at a higher conceptual plane.

2.3. Transformations of a geodatabase

The task of polygon overlay described in Section 2.1 can be seen as a transformation of a geodatabase G into a new geodatabase G' [10]. In this case, the effect is to add a new class of polygon features, by combining two existing classes, giving the a new set of polygons concatenated attributes, and creating boundaries by reassembling fragments of the input boundaries. Since we now have formal and rigorous models of the set of possible GIS data types (see, for example, the OGC Simple Feature Model), it should be possible to list every possible type of transformation, in other words to enumerate all of the possible GIS operations. It would be necessary to add operations that do not modify the database but instead result in numerical or tabular output, or in visualizations. Nevertheless research along these lines might yield a formal and provably complete structure for GIS functionality.

3. Spatial Thinking

One place to look for ideas about reorganizing the user interface would be the textbooks of spatial analysis, since they also need conceptual frameworks for dividing the body of material into chapters and ordering the chapters. Bailey and Gatrell [17] organized their text by data type, describing the forms of analysis that could be conducted on sets of points, sets of polygons, etc. This is helpful, but it does not reflect the fundamental distinction between continuous fields and discrete objects, since a single set of points or a single set of polygons can represent either, but the operations that can be performed are entirely different. For example, users often confuse the operation of kernel density estimation, which can be applied to mapping the density of disease mortality, for example, with spatial interpolation, which is used to interpolate values of a variable such as temperature between the locations where it has been measured. The former case is appropriate for creating a surface of the density of discrete objects, while the latter is used to interpolate a continuous field. Applying kernel density estimation to points that sample a field, or spatial interpolation to discrete objects, makes no sense. Yet GIS software typically does not prevent the user from doing so, seeing both cases as collections of point features.

Mitchell’s books on spatial analysis ([18], [19]) use a different approach in which analysis is organized according to the concepts and questions that motivate a user to use GIS. Volume 1, for example, includes the following chapters: Mapping Where Things Are, Mapping the Most and Least, Mapping Density, Finding What’s Inside, Finding What’s Nearby, and Mapping Change. These titles reflect the GIS-independent thoughts in the mind of the user, including well-understood terms such as “inside” rather than the technical language of GIS functionality. The user is not concerned with the process of turning such thoughts into sequences of GIS functions, but only with the purpose that GIS technology is in principle designed to serve. Section 2 presented three bases on which GIS functions might be simplified. In this section I explore a broader notion, that one might organize the functionality of GIS by the spatial concepts that already exist in the mind of a user.

Recently there have been several efforts to enumerate such concepts, although the broader literature extends back for many decades. At the Center for Spatial Studies at the University of California, Santa Barbara, we have developed a set of Web resources (teachspatial.org) designed to sythesize this literature, integrate it into a defined set of spatial concepts, create a number of organizing structures that allow the set to be navigated, and develop teaching resources that can elaborate the most important of the concepts. The resources are largely the work of Karl Grossner, with assistance from the author, Donald Janelle, and Josh Bader. To date we have found some 185 concepts, distributed across the literatures of many disciplines.

Besides alphabetical order, perhaps the most obvious way to organize the concepts is through metrics of similarity, bearing in mind that the same concept may appear in different forms and under different names in several disciplines. There may also be hierarchical part-whole relationships, if one concept is a subset of another. The side also allows concepts to be accessed by discipline, in an effort to identify the disciplines that most often use each concept. We have also prepared various short lists of the concepts most likely to be employed in GIS analysis (see, for example, [20], Chapter 2).

The next stage of this effort will be the development of a new interface to the ArcGIS Toolbox, that shields the user from the technical terminology of GIS and allows interaction through the spatial concepts that each function addresses. At the same time the approach will employ the ideas developed in Section 2, to reduce the technical level of interaction to a more fundamental set of operations.

4. Conclusion

This paper has identified a major problem in GIS—that after four decades of development, we lack anything approaching a standard, formalized functionality that can support SOA, and that integrates well with the thought processes of the user. As a result, while great progress has been made at achieving interoperability of data, almost no progress has been made on functional interoperability. The GIS user interface remains complex, hard to learn and use, and lacking in any consistent conceptual or theoretical framework.

I hope that this paper has stimulated interest in a topic that is intimately related to spatial thinking and GIS. We need a concerted, collaborative effort to solve this problem if GIS is to be as easily and intuitively accessible as it should be.

Acknowledgements

The ideas expressed in this paper have benefited from discussion with numerous individuals interested in spatial thinking and GIS. I am especially grateful to Karl Grossner for his work on teachspatial.org and to the participants at a workshop in Santa Barbara in February 2011 on Formalizing GIS Functionality: Shaowen Wang, Jochen Albrecht, Naicong Li, Linna Li, Wenwen Li, Josh Bader, and Xuan Shi. The workshop was funded in part by the National Science Foundation, award OCI 1047916.

References

[1] Goodchild MF. A spatial analytic perspective on geographical information systems. *International Journal of Geographical Information Systems* 1988; **1**:327–334.

[2] Tomlin CD. *Geographic information systems and cartographic modeling*. Englewood Cliffs, NJ: Prentice Hall; 1988.

[3] Ritter GX, Wilson JN. *Handbook of computer vision algorithms in image algebra.* 2nd ed. Boca Raton: CRC Press; 2001.

[4] van Deursen WPA. *Geographical information systems and dynamic models: Development and application of a prototype spatial modelling language.* Utrecht: Koninklijk Nederlands Aardrijkskundig Genntschap/Faculteit Ruimtelijke Wetenschappen Universiteit Utrecht; 1995.

[5] Takeyama M, Couclelis H. Map dynamics: Integrating cellular automata and GIS through geo-algebra. *International Journal of Geographical Information Science* 1997; **11**(1):73–91.

[6] Berry JK. Fundamental operations in computer-assisted map analysis. *International Journal of Geographical Information Systems* 1987;1:119–136.

[7] Dangermond J. A classification of software components commonly used in geographic information systems. In: Peuquet DJ, O'Callaghan J, editors. *Design and implementation of computer-based geographic information systems*. Amherst, NY: International Geographical Union, Commission on Geographical Data Sensing and Processing; 1983, p. 70–91.

[8] Maguire DJ, Dangermond J. The functionality of GIS. In: Maguire DJ, Goodchild MF, Rhind DW, editors. *Geographical Information Systems: Principles and Applications* 1: 319-335. Harlow, UK: Longman Scientific & Technical.

[9] Rhind DW, Green NPA. Design of a geographical information system for a heterogeneous scientific community. *International Journal of Geographical Information Systems* 1988; **2**(2):171–190.

[10] Goodchild MF. Towards an enumeration and classification of GIS functions. In: *Proceedings, International Geographic Information Systems (IGIS) Symposium: The Research Agenda* Vol II. Washington, DC: National Aeronautics and Space Administration; 1988, p. 67–77.

[11] Longley PA, Goodchild MF, Maguire DJ, Rhind DW. *Geographic information systems and science*. 3rd ed. Hoboken, NJ: Wiley; 2010.

[12] Zeiler M. *Modeling our world: The ESRI guide to geodatabase design*. Redlands, CA: ESRI Press; 1999.

[13] Burrough PA. Development of intelligent geographical information systems. *International Journal of Geographical Information Systems* 1992; **6**(1):1–12.

[14] Smith B, Varzi AC. Fiat and bona fide boundaries. *Philosophy and Phenomenological Research* 2000; **60**:401–420.

[15] Kemp KK. Fields as a framework for integrating GIS and environmental process models. Part one: Representing spatial continuity. *Transactions in GIS* 1997; **1**(3):219–234.

[16] Kemp KK. Fields as a framework for integrating GIS and environmental process models. Part two: Specifying field variables. *Transactions in GIS* 1997; **1**(3):235–246.

[17] Bailey TC, Gatrell AC. *Interactive spatial sata analysis*. Harlow, UK: Longman Scientific and Technical; 1995.

[18] Mitchell A. *The ESRI guide to GIS analysis. I. Geographic patterns and relationships*. Redlands, CA: ESRI Press; 1995.

[19] Mitchell A. *The ESRI guide to GIS analysis. II. Spatial measurements and statistics*. Redlands, CA: ESRI Press; 2005.

[20] De Smith MJ, Goodchild MF, Longley PA. *Geospatial analysis: A comprehensive guide to principles, techniques and software tools*. 3rd ed. Winchelsea, UK: The Winchelsea Press; 2009.