Requirements for the development of computer-based urban information systems

by STEVEN B. LIPNER

The Massachusetts Institute of Technology
Cambridge, Massachusetts

INTRODUCTION

Since early in this decade urban planners and systems analysts have advocated the development of computer-based urban information systems. Such systems would store detailed data about the environment in which planning agencies and governments operate. They would be organized to lend integration to data from diverse sources, to provide quick preparation of reports and to simplify and automate numerous clerical functions. Many attempts have been made to develop urban information systems with the characteristics mentioned above. Most have been unsuccessful[1] for a combination of technical and organizational reasons. This paper considers some technical requirements for planning information systems which deal with data associated with urban locations. The requirements are developed on the basis of experience in providing a prototype urban information system to the Boston Model Cities program. The next section describes briefly the experience of providing an information system to the Boston Model Cities program. Succeeding sections draw on this experience to develop general technical requirements for urban information systems. A technique for aggregating data by geographic area is presented and its implications for system file structure and utilization are explored.

Information system for the Boston Model Cities Administration

During the spring of 1968, M.I.T. staff members held a number of meetings with members of the staff of the Boston Model Cities Administration to determine how M.I.T. might assist Boston's Model Cities program. One of the major desires of the Model Cities staff members was to see if an urban information system could be used to aid their planning and program evaluation activities. The Model Cities Administration was undertaking a survey which would determine the land use, building condition, and building size associated with each parcel in the Model Neighborhood Area. It was agreed that this data would make an acceptable basis for a prototype urban information system. Model Neighborhood residents employed by the Model Cities Administration were trained in key-punching and prepared approximately 8000 cards, one for each parcel in the area. (For comparison, the city of Boston contains about 100,000 parcels.)

The survey data was input to the ADMINS[2,3] system operating on the time-shared 7094[4] at the MIT Computation Center. ADMINS is an interactive program capable of performing data selection and cross-tabulation. It was designed for use in the analysis of social science surveys, and is best suited to operating on small files of coded or integer-valued data items. It is weakest in the areas of data modification, large file handling, and real or alphanumeric data manipulation.

Initial preparation of the data for ADMINS analysis was judged too complicated and machine-oriented a task to be performed by persons with little computer training. Accordingly the data was prepared for analysis by MIT personnel experienced in programming and in the use of ADMINS. The data preparation was simplified by the ability of ADMINS to accept data in arbitrary codes and formats and by the interactive mode in which it is used. Errors in the data were reported by ADMINS programs and corrected by using the time-sharing system's general purpose editing capabilities to modify the input files.

The analysis of the Model Cities survey data was performed by three groups of people: MIT staff

members with substantial computer experience, professional urban planners with little or no prior computer experience, and Model Neighborhood residents with neither computer experience nor extensive formal education. All three groups easily mastered the mechanics of producing desired cross-tabulations, although a natural "fear" of the computer terminal had to be overcome by those new to it.

The response of the planners to the prototype urban information system was both interesting and significant. Although they had been instructed in the use of ADMINS at the terminal, and given freedom to produce reports as needed, the planners preferred to contact MIT or Model Neighborhood personnel, describe verbally the required tables, and have the resulting hard copy delivered to them. Whether this phenomenon was caused by the lack of proximity of the planners to the terminal, by the relatively tedious ADMINS language, or by a basic reluctance of planners to use the computer directly remains undetermined. (Placement of a terminal at the Model Cities office has been planned for some months but has been delayed by various administrative and operational problems.) When the planners have more direct access to a terminal and are provided with a system which, unlike ADMINS, is designed to serve as a true urban information system, it should be possible to determine if experienced planners without computer experience can successfully be trained and encouraged to use a computer as a planning tool. The implications of such a determination are discussed in the next section.

The analytic results produced for the planners using ADMINS were useful, and all agreed that they were pleased with the results of the analysis. The limited computer experience, however, whetted the planners' appetites for more diverse capabilities. These capabilities included:

1. The ability to aggregate data by arbitrary geographic areas such as school districts, without being required to list explicitly every block contained in each area.
2. The ability to produce maps and graphs as well as tables.
3. The ability to merge data gathered by operating agencies and survey research organizations with stored data.
4. More general capabilities for numeric and alphabetic data processing than those provided by ADMINS.

The experiment in computer-aided Model Cities planning has been successful in two senses. First, it provided valuable insights into the capabilities required of an urban planning information system. Second, it introduced a group of planners to computer-aided analysis. In the future these planners should provide valuable data on the mode of man-machine communication appropriate for an urban planning information system.

Requirements for urban information systems

The experimental provision of computer support to planners described in the previous section provided several insights into the capabilities required of an urban information system and the specific features required to implement them. Perhaps the most important capability indicated is that of combining and using in a single information system data from a variety of sources. Special surveys are an expensive and short-lived source of planning data when compared with operational data which must be maintained, often in machine-readable form, by agencies other than the planning department. Operational data from a given agency, in order to be useful to the planner, must be combined with planning survey data and often with data from other public or private operational agencies. Since different agencies often use different identifiers for each parcel, and since the street address is the only common and (presumably) unique parcel identifier, the conclusion is reached that a useful planning information system must deal with parcels identified by street address. Address matching programs[5] have been developed which standardize the formats of street addresses keypunched in free format. They must be included in an urban information system, along with file structures appropriate for the identification of parcels by street address. The need to merge data from differing sources implies the possibility of varying amounts of data describing a single parcel. Such possibilities must be handled by a flexible but efficient data file structure.

A second major requirement of an urban information system is the ability to aggregate parcel data by arbitrary geographic area. This ability is especially important in view of the numerous overlapping administrative and planning districts into which urban areas are divided. Programs have been developed[6,7] which aggregate data into districts by first assigning coordinates to each parcel, and then testing each parcel to see if its coordinates lie within a district. Such programs work but seem suited mainly to sequential storage systems using fast computers. The reasons for this observation and an alternate technique based on street addresses will be presented in the next section.

The importance of graphical display of data to planners was emphasized during the initial work with model cities planners. Any really useful urban infor-

mation system must produce graphical as well as tabular output, preferably with minimal user description of coordinates, scales, etc. Existing programs and systems[8] are capable of producing a wide variety of graphic outputs. The major problems in applying these to urban information systems are, first, assuring that the outputs they produce are those required by planners and second, integrating the graphic components with data management components to minimize the complexity and cost of producing the outputs.

The area of man-machine communication is one which may be critical to the success of urban planning information system design. The experiment described above produced results which can only be described as inconclusive. However experience in the use of computers by engineers[9] would seem to indicate that the use of computers by persons who are not computer-oriented is greatly aided by the availability of interactive problem-oriented languages. In order to produce definitive results in the area of communication between computer and planner it will be necessary to provide both better terminal access and a problem-oriented language superior in both power and usability to that of ADMINS. The growing presence of planners who have had computer training should provide further assistance in improving man-machine communications.

In re-examining the requirements developed in this section, we find that all except those of geographic aggregation of data, address matching and graphical output would be common to any powerful information system: file structures which allow items to be described by varying numbers of attributes, file structures for rapid data retrieval, and powerful problem-oriented retrieval languages are all provided by many modern information systems.[10,11] Of the required features which appear unique to urban information systems the most significant seems to be that of geographic aggregation of data. Address matching is essentially a preprocessor function and graphic output an important output processor, while the geographic aggregation method will have a significant effect on the cost of many retrieval requests and some influence on internal file organization. For this reason, the next section is devoted to a brief description of an alternative to existing schemes for geographic aggregation of data.

A technique for geographic aggregation of parcel data

The problem of geographic aggregation of parcel data in urban information systems has typically been handled by "point-in-polygon" programs.[6,7] Such programs require that each parcel which is included in the information system be identified by its x-y coordinates. An area for which data is to be aggregated is described as a polygon by specifying the coordinates of its vertices. Each stored parcel is tested by counting the intersections of a ray of arbitrary direction originating at its identifying point with the sides of the polygon. If the count is even, the point (and hence the parcel) is outside the polygon. If the count is odd, the point is inside (Figure 1).

Although the point-in-polygon test is a workable technique for geographic aggregation of data, it poses two problems. First, and less significant is the problem af assigning coordinates to every parcel. This problem is easily solved by representing every street as a sequence of line segments and using the numerical value of each parcel's address first to select the segment containing the parcel and then to define the parcel's coordinates by interpolation between the segment's end points. The second and more serious problem presented by the point-in-polygon technique involves processing time. Since the point-in-polygon technique is a test on one parcel, every parcel recorded by a system must be tested to determine which parcels should be aggregated into a given area. Thus, the technique is ill-suited to systems employing direct-access storage devices which could allow selective access to desired parcel data. Furthermore, the calculations required to determine whether or not each parcel lies in a given area involve one line intersection for each side of the area. On some small computers this calculation may be relatively time-consuming. Thus even if the parcel data base were recorded on tape, the time required to select those parcels in an area could be governed by processing time rather than by the time required to move and read the tape.

Techniques have been suggested[12,13] which, by dividing an urban area into subareas, would reduce the sequential file searching required by the point-in-polygon algorithm. These techniques would require checking of the retrieval area for overlap with pre-established subareas before individual parcels in the

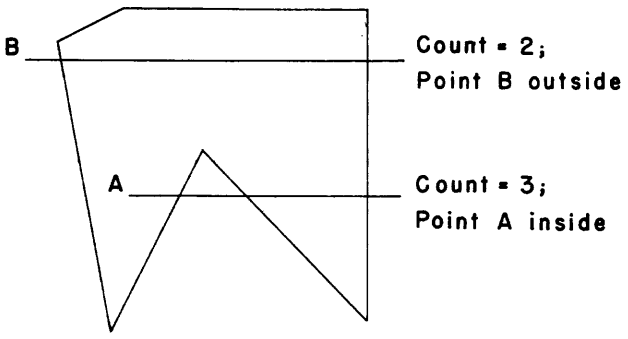

Figure 1—Point-in-polygon test

subareas were examined. If the check showed no overlap, no further examination of the subarea would be required. Otherwise every parcel in the subarea would be checked. The disadvantages of this method are principally associated with the size of subareas. A large number of small subareas requires a large number of overlap tests, while if a small number of larger subareas are used, there will be a large number of parcels requiring point-in-polygon testing included in each selected subarea.

An alternative to the point-in-polygon technique for the geographic aggregation of parcel data was suggested first by Farnsworth[14] and later proposed independently and in more detail by Parsons.[15] The algorithm involves using a map of the street network of the urban area within which new geographic areas are defined. Given a list of the names of the streets surrounding the area of interest, the algorithm produces a list of those parcels within the area. The paragraphs below present an illustration of the algorithm, followed by comments on the map file structure required to implement it.

In considering the map of Figure 2, let us assume we wish to isolate the area bounded by streets A, H, D and E. We first scan the street A until we locate the set of street segments (portions of a street between two intersections) on it between E and H. We then scan street H, marking the segments between A and D, street D for the segments between H and E, and street E for the segments between D and A. Since the list of bounding streets was given in a clockwise direction, we know that blocks inside of the desired area are to its right. If we have recorded the numbers of the blocks to the right and left of each segment, seen facing in the direction of increasing addresses, we may now isolate those blocks inside the bounding streets. To do this we record blocks to the right of

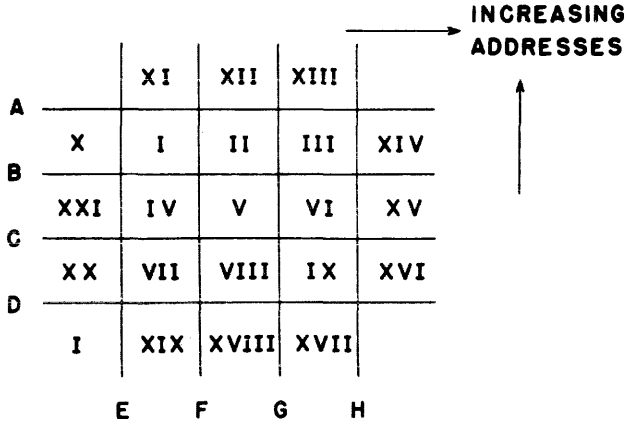

Figure 2—Map for geographical retrieval

segments whose increasing address direction coincides with the direction of the area boundary (street A and E) and blocks to the left of segments whose addresses run opposite to the boundary (streets D and H). Applying this procedure we obtain the list of contained blocks in Figure 3.

As we make the list of contained blocks, we may also make a list of non-contained blocks (Figure 4). These are blocks opposite the contained ones which lie just outside (to the left) of the area boundary. Now we may make a list of blocks adjacent to those blocks listed in Figure 3, excluding blocks already listed as contained or non-contained. This list contains only one block, block V. Enumerating the blocks adjacent to block V we find that all have already been listed as contained. Thus all blocks within the area of interest have been isolated. From the list of blocks in the area, we may develop a list of the address ranges along contained streets or of the parcel numbers of parcels contained in the area.

The algorithm and problem described are reliable only when used with a street network in which no two streets intersect more than once. Techniques have been developed by the author which generalize the algorithm to handle cases in which two streets may intersect more than once, by eliminating resolvable ambiguities or by reporting the presence of irresolvable ones. The generalization requires changing the initial analysis of the list of streets bounding the area from a one-pass to a multiple-pass operation. The first pass isolates all possible sequences of segments which could surround the desired area. The second and succeeding passes eliminate incorrect paths by searching for discontinuities in the transitions from one street to the next. The process is continued until one correct path remains or until no further incorrect ones can be detected.

Two files are used to allow a computer program to implement the algorithm described above. The first contains data about street segments for every street in the map, while the second contains lists of the blocks

I, II, III, VI, IX, VIII, VII, IV

Figure 3—First list of contained blocks

XI, XII, XIII, XIV, XV, XVI

XVII, XVIII, XIX, XX, XXI, X

Figure 4—List of non-contained blocks

adjacent to every block in the map. The first file is used to isolate the sets of blocks just inside and outside an area described by its bounding streets. The segments along a street are ordered by increasing address range, and each segment is described by left and right block numbers, beginning and ending node numbers, and intersecting streets. Additional data on street address ranges and node coordinates for each street are typically included to broaden the utility of the segment file. The block file must include the numbers of the blocks surrounding each block, and should contain data to allow conversion from the numbers of the blocks in the desired area to the data themselves—either as street names and address ranges, as parcel numbers, or as disk identifiers of data records. Both files described above may be produced as by-products of the DIME editing technique[16] described by Cooke and Maxfield.

The algorithm outlined above for using a street network to facilitate geographic aggregation of parcel data has both advantages and disadvantages when compared to the point-in-polygon technique. Its principal advantage is that it is essentially a direct-access technique. The time required to isolate the identifiers of those parcels in an area is proportional to the number and length of the streets surrounding the area and to the number and complexity (number of adjacent blocks) of blocks in the area. Small areas may be isolated very quickly. If some sort of direct-access storage is used for parcel data, the parcels in the area are the only ones retrieved. If sequential storage is used, the algorithm can at least produce a list of parcel identifiers (for example address ranges) which will allow much speedier checking of individual parcels than would be the case with the point-in-polygon routine. The principal disadvantage of the street network technique is its limited flexibility. While the point-in-polygon technique may be used to select parcels in any area, the network technique is clearly applicable only to areas made up of whole blocks. This problem is potentially most serious in analyzing areas such as new highway corridors which do not follow block boundaries. It seems possible that performing such analysis by using the point-in-polygon technique on a set of parcels selected by the network technique might be more economical than applying it to all parcels in a city. However, this hypothesis must be verified.

File structure

The basic implication of the geographic aggregation technique proposed above is that a direct-access file system is very desirable. The principal requirement of this structure is that it be capable of being tied to the block data of the street map file. One flexible way of establishing this tie is to use street address as the major identifier of each parcel and to store street names (or identification numbers) and address ranges in the block file of the street map. The street names and address ranges defining all block faces (one side of a segment) in an area could be merged together and sorted into an order corresponding to that of the parcel data file. Then retrieval from the parcel file could be directed by the sorted output of the aggregation algorithm. Retrieval of data about those parcels in a given area could proceed at a speed governed only by the efficiency of the parcel file's indexing scheme. Variable amounts of data for a single parcel could be stored either in variable-length data records or in multiple files each using street address as primary identifier. Two major advantages of using street address as the primary parcel identifier are, first that all inquiries about parcels by street address would be facilitated and, second, that additions or deletions of occupied addresses within a block face necessitate no alterations to the network data describing that block face.

If a sequential file structure is to be used for parcel data, for reasons of restricted data access, economy, or data volume, the comments about using street address as primary identifier still apply. Although sequential processing becomes imperative, the simplicity of processing allowed by using street address ranges as output from the geographic aggregation algorithm will still minimize the actual processing time required to select parcel data. This minimization may be important when processing data on a small machine or in a partition of a large one.

A planned experimental system

The techniques used above are to be put into practice in an experimental information system for use by the Boston Model Cities Administration and MIT Urban Systems Laboratory. The system will include a street network file and street network geographic aggregation algorithm. The street network file will be tied to a parcel data file by street addresses. Multiple parcel data files will be used to handle multiple data sets (initially housing survey and demographic survey files) on direct-access storage. Control and problem-oriented language facilities will be provided by the ICES system.[17] The system should be implemented by June, 1969 and will be operated as a planning aid for the Model Cities Administration by Model Cities and MIT staff members. In addition to providing basic statistical and cross-tabulation facilities, it is hoped

that the system will allow the addition of analytic and modelling capabilities by planning researchers.

ACKNOWLEDGMENT

The work reported here was aided and influenced by many people over the last year. Especially worthy of mention are Professor Charles Miller, Professor Robert Logcher, Mr. William Parsons, Mr. Ronald Walter, Mr. Donald Cooke, and Miss Betsy Schumacker of M.I.T., Mr. Edward Teitcher and Mrs. Colette Goodman of the Boston Redevelopment Authority, and Mr. Michael Warren, Mr. Richard Harris, Mr. Samuel Thompson, and Mr. John Myers of the Boston Model Cities Administration. The work reported herein was conducted at and sponsored in part by the Urban Systems Laboratory of the Massachusetts Institute of Technology.

BIBLIOGRAPHY

1 O E DIAL
Urban information systems: A bibliographic essay
Urban Systems Laboratory M I T 1968
2 S McINTOSH D GRIFFEL
The ADMINS primer
Center for International Studies M I T
3 S McINTOSH D GRIFFEL
The language of ADMINS
Center for International Studies M I T
4 P A CRISMAN
The compatible time-sharing system: A programmer's guide
M I T Press
5 H H COCHRAN
Address matching by computer
Proc Sixth URISA Conference 1968
6 R B DIAL
Street address conversion program
Urban Data Center University of Washington
7 S NORBECK B RYSTEDT
Computer cartography point-in-polygon programs
BIT 7 1967
8 D F COOKE
Systems, geocoding and mapping
Proc Sixth URISA Conference 1968
9 C L MILLER
Man-machine communications in civil engineering
Department of Civil Engineering M I T
10 R E BLEIER
Treating hierarchical data structures in the SDC time-shared data management systems (TDMS)
Proc A C M National Conference 1967
11 E W FRANKS
A data management system for time-shared file processing using a cross-index file and self-defining entries
Proc S J C C 1966
12 K J DUEKER
Spatial data systems
Northwestern University
13 S B LIPNER
File structures for urban information systems
Internal Working Document M I T 1968
14 G L FARNSWORTH
Contiguity analysis using census data
Proc Fifth Annual URISA Conference 1967
15 W A PARSONS
Unpublished class project report
M I T Subject 1 152 1968
16 D F COOKE W H MAXFIELD
The development of a geographic base file and its uses for mapping
Proc Fifth Annual URISA Conference 1967
17 D ROOS
ICES system: General description
Department of Civil Engineering M I T