FILOMAT (Niš) 9:2 (1995), 357–366

Filomat '94, Niš, October 22-24, 1994
Geometry. Computer Sciences

A SYSTEM FOR STORAGE, MANIPULATION AND CONTROL OF DIFFERENT GRAPHICS FORMATS

Zoran Putnik

ABSTRACT. *In this paper, a detailed outline of a system for memorizing, manipulation and control of pictures given in different graphic formats is given. System consists of several modules, already known and available, but the value of the system is mainly in combination of several useful functions, enabling complete and efficient management of miscellaneous kinds of pictures and cutting on expenses and possible errors in manipulation with various graphics formats.*

1. Introduction

Manipulation of drawings and other graphics elements is mush more then just a simple storing/retrieving of data and drawings. It is rather a complicated process of drawings' creation - starting by a designer, external and internal skilled consultants, through artists who actually make drawing, up to users of the finished drawings, or some of its parts. During drawing creation, standard parts from shared or private libraries are incorporated, or referred to, and usual necessary data - names, dates, references, are given. Dates of drawing creation are still not the final dates of need for a drawing. Often changes, especially for technical drawings, demand easy access to a drawing for a long period. This demand, naturally includes a need for some tools for transferring drawings from one graphics format to another. As for any other data stored in a computer, manipulation with drawings requires handling of standard problems: efficient storing system, fast and simple data retrieval, enabling changes in existing drawings or using existing drawing in creation of a new ones, managing an efficient data base about drawings and related data, transferring drawings from paper to a computer and similar.

1991 *Mathematics Subject Classification.* 68U05, 68P15.

Work supported in part by Ministry for Science of Republic Serbia

Toward overcoming of mentioned problems, in this paper, a system for storage, manipulation and control of drawings in different graphics formats is given. Separate modules, this system consists of, are not new nor original, instead, most of them are available for a commercial use in some form. Value of this system is mainly in unifying and combining all necessary functions, enabling simple and efficient control of data and drawings flow even for a long period and reducing expenses and chance for errors in drawings' manipulation.

2. State of the art

We can notice several different logical modules in modern systems for information and documents management (from now on CDMS - Corporate Document Management Systems):

- module for storing information - data base
- module for data search - key-words data base
- module for documentation viewing and control
- communication module - fax, e-mail, modems
- module for controlling computer network
- module for changing documents (in original programs)
- module for automatic text recognition
- module for handling pictures

Separate programs for each of the mentioned modules are developing for years (more or less successfully), everything toward creation of "paperless office." Information stored in digital - electronic form, does not need a paper as a storage media. But, to be easily available to the user, it demands another elements of a system for data storing and retrieval. As main elements, we can mention:

- computer for data storage - "main computer"
- (computer for communication with a "main computer")
- software for "reading" and "presenting" given information

As much as textual data are concerned, several mostly used text-processors can be identified, that each CDMS have to support, with always present, final solution, of recognizing text in its simplest form - ASCII standard. For graphics data, such standards do not exist. We can talk of "most frequent" graphics forms, i.e. *.PCX, *.TIF, *.GIF, *.IMG - as bit-mapped, or *.DWG, *.DXF or *.CDR - as vectorized, but, basic standard does not exist jet.

Computer system for storage, manipulation and control of graphics formats is very important subsystem of a system for creation, management and archiving documents - CDMS. It must successfully and efficiently integrate some commonly accepted programs for graphical documentation management, from scanning and character recognition programs, through programs for editing bit-mapped, vectorized or ASCII graphics files and transferring drawings from one form to another, up to programs for presentation and printout graphical documentation on various kinds of output devices. This subsystem is also an useful step towards creation of a multimedial data base, which will enable fast and simple finding, retrieval and exporting any document stored in any existing form. Modular and flexible, this subsystem has to be (theoretically) usable equally in small and in big business systems, despite working area. In its nature, such subsystem assumes (and gives best results) computers connected in a network, which again permit successful control of document flow, transforming several "personal" computers in an efficient information system.

A system for storage, manipulation and control of graphics formats (somewhere called EDM, standing for "Engineering Drawing Management" or "Electronic Data Mng" or "Engineering Data Mng") has to emulate, for a successful work all standard activities in a process of creation, storing and "maintenance" of drawings. Main of these activities are:

- Control function - **DIRECTOR** - a module handling work and communication of other modules, controlling them and controlling users' behavior according to his priority level;
- Data storage function - **LIBRARIAN** - a relational data base, enabling a search for a specified drawing based on a key-word and creation of a report on a drawing including place and time of creation, author, dates and types of changes, current status, list of keywords and list of access rights;
- Digitalization function - **SCANNER** - a module for connection between "old" and "new" technology of drawing creation and for connection with third parties, which produce their drawings in a paper form. It should also contain some standard way of data compression (for example - scanned drawing of A0 format, with resolution of only 400 dpi, as a result requires 40 MB of storage space if stored in a bit-mapped form);
- Editing function - **EDITOR** - a module that enables that drawing we want to change (coming through module LIBRARIAN or module SCANNER) can be edited either with standard geometric functions (scaling, rotation, translation ...) or manually (adding or deleting picture pieces, coloring, text editing...)
- Vectorization function - **VECTOR** - a module that (if needed) enables transformation of bit-mapped drawing into a vectorized drawing. Experience shows that this function is not always necessary,

since very good abilities for changing raster images are developed, and on the other hand, vectorization process takes a lot of time not always bringing significant improvement of quality;

- External communication function - **TRANSFER** - a module that overcomes a problem of using different software for drawing creation and enables combining of drawings created on different places in a different ways;
- Text recognition function - **READER** - a module enabling usage of text documents created in most standard text processing programs, or, if nothing else is possible, enables text recognition using usual optical character recognition techniques;
- Viewing and printing function - **OUTPUT** - a module enabling that library drawings, can be viewed on (any kind of) a screen, and/or plotted/printed on (any kind of) a printing device.

3. System modules

3.1. Drawings storage module - data base.

Creation of a complex drawing, consisted of several drawings, sometimes already created in different graphics formats, using different software tools, is usual very difficult. Reason for this is existence of three principally different formats - bit-mapped drawings, vectorized drawings and drawings created of ASCII characters - with a huge number of subtypes for the first two. Emerging of a new version of existing graphics software, usually brings lots of problems to the end-users. Besides that, for each graphics document, some extra information is needed, for example: date and time of creation and editings, names of authors, coauthors, consultants and "maintenance" employees, references to parts taken from standard libraries or to bigger drawings of which the given one is a part of, and so on. The most convenient method for storage of this kind of data is some standard, relational data base, which will enable easy sorting, searching and editing of existing data.

This module has to provide a simple and obvious searching method through the graphical data base on any criteria, without previous knowledge of programming languages or data bases. This can be achieved through a simple and readable graphics interface, enabling easy entering of wanted search criteria. Multiple criteria search, easy access to the results of a previous search and other similar, practical options are usual in any serious data base, so there is no need to explain them separately.

As a first result, a search gives simplified, smaller picture of all drawings satisfying given criteria. Later, those pictures, depending on users access level, could be viewed, edited, printed, commented and so on. Naturally, for advanced users, it would be very useful to have programming language, which can define either aestethic (i.e., shape of a search screen) or essential search details (definition of new fields of a data base, with their attributes, creation and organization of an archive of technical and business documentation, catalogues of products, data bases of persons involved in drawings creation and so on).

3.2. Module for transferring paper documentation into electronic form.

Because of paper documentation inherited from previous work and because of need for cooperation with other parties producing paper drawings, this module is necessary in any system for storage, manipulation and control of graphics formats. It should cover following functions:

- picture scanning
- editing of errors of scanning
- optical character recognition
- editing of bit-mapped pictures
- picture vectorization
- editing of vectorized pictures
- editing of ASCII pictures

3.2.1. Subsystem for scanning.

Any "real-life" business system, besides documentation created on a computer, is doomed to have contact with paper documentation. That documentation is, seldom or rarely, used, saving of some documentation is usually legal obligation. Transformation of that documentation into an electronic form by repeated drawing is usually too complicated and too expensive. Instead, it is more natural to keep it in a computer archive in a form of scanned pictures. After scanning, these pictures can be edited more or less, vectorized, if necessary, or transformed into text, which all are parts of subsystems that will be mentioned later.

Process of scanning and editing of scanned pictures, should be, according to latest trends in this field [5], equipped with tools for performing following functions:

- scanning errors' correction
- straightening of aslanted pictures
- removal of "snow" emerging because of a dirt on a paper
- thickening or thinning of lines
- definitions of separate, different filters, for specific parts of a picture
- linking of disconnected contours, or separating of badly connected contours
- standard functions for adding, editing and deleting parts of drawing

3.2.2. OCR subsystem.

This module confirms to all standard demands for this class of programs, which will not be especially discussed in this context. It should only be emphasized that this subsystem has to supply a connection between a graphics document in an unknown format and ASCII file obtained by process of scanning and optical character recognition. This is, naturally, performed only as a final measure, if information about the contents of a picture cannot be obtained by any other means.

3.2.3. Subsystem for picture editing.

This is again a standard subsystem, that should not be explained in much details. It should be only mentioned, that this subsystem in fact is consisted of three separate parts, for editing different types of drawings - bit-mapped, vectorized and ASCII character drawings. Since one drawing can be created as a combination of all these types, all editing tools have to be available at any moment.

3.2.4. Vectorization subsystem.

There is often a need for large amount of changes that should be performed on an existing drawing. This is usually much easier (end with higher quality) performed on a vectorized picture. Besides, vectorized picture, compared to a bit-mapped picture, usually take much less space, which is a very important demand in this field,. Considering all mentioned, subsystem for vectorization is an obligatory part of a system for storage, manipulation and control of graphics formats.

There is a set of standard tools for this process and usual procedures for manual and automatic vectorization. Here, some more advanced actions about vectorization will be underscored:

- definition of vectorization "filter" (for example artistic or technical, or even more specific - electronic, architect, engineering ...), which as a consequence, brings different definitions of some standard vectorization parameters:
  (1) characteristics for approximation of curves,
  (2) definition of smallest object that is vectorized,
  (3) minimal offset of horizontal/vertical line that is not neglected,
  (4) method of text recognition,
  (5) minimal distance that separates two lines and so on.;

- enabling manual or automatic vectorization and vectorization of a whole picture, of a part of a picture or definition of a part of a picture that should not be vectorized;

- enabling recognition of at least some basic contours - circle, ellipse, square, for example - as contour, and not as a combination of simple lines and curves. The same should stand for a combination of those basic shapes. For example, a square written *IN* a circle, should be vectorized as those two contours, and not as a combination of four lines and four curves;
- text, as a part of a picture, can be recognized either as graphics (transformed to curves), as letters (i.e., optical character recognition of ASCII characters) or completely removed from a drawing;
- after finished automatic vectorization, there should be a possibility for comparison of bit-mapped original and achieved result. Naturally, there should be an ability for additional, manual changing of vectorized picture;

3.3. Output module.

This module has to enable rough and/or detailed view of "all" important graphics formats - bit-mapped or vector, including documents created by important text-processing programs, spreadsheets or data-base programs. For this module, only a quick and simple access to document is important, including output abilities on all output devices, screens, printers and plotters. Eventual changes of documents should not be incorporated into this module, since these abilities are a part of another modules.

3.4. Module for manipulation of technical drawings.

Special problem in this field is production, maintenance and editing of technical drawings. During creation, technical drawings go through many phases of treatment, addition and editing, so that, as a result there is too many paper versions of a drawing, usually right one at the wrong place. Chief problem with technical drawings (for example drawings of bridges, buildings and similar) is that they have to be saved and maintained for several tenths of years.

Unfortunately, introducing computer aided design (CAD) into this area, can put us in an even worse situation. Part of documentation is saved on a computer, part on a paper, some initial versions of a drawing are declared final, while some final versions are rejected as unnecessary. In order to overcome these problems it is urgent to, right after introduction of computer aided design, transfer all documentation into electronic form, no matter of what kind, origin or shape they are and organize a data base to accompanies that documentation. Later phase will usually demand several computers connected into network.

3.5. Communication module.

This module has to enable safe, fast and easy communication between different modules in a system for storage, manipulation and control of graphics formats. As much as an user is concerned, it should supply simple usage, different methods to perform functions (keyboard, mouse, arrows ...), readability of a screen, easy-to-use help system and all other standard requirements for a proper user-friendly graphic interface [7].

4. Future development

It seems, considering fast development of science, especially computer science, that it will be possible in near future to spread system like this one in several different areas. Even though commercial versions are still unavailable, some fields are developing very fast and we can expect soon expansion of system for storage, manipulation and control of graphics formats, for example with:

(1) Optical recognition of text - not characters

Latest research shows [1] [4], that optical character recognition systems are very close to their upper limits. Although those limits are rather high (over 95%), for large texts, and, more important, for texts that allow no errors, this is insufficient. Consequently, organizations that want to work with "electronic documents" cannot rely on them. These facts, initiated research in a field of optical recognition of texts, based on analysis of a document structure and its contents. A system like that, must contain several text characteristics: big dictionaries, text styles, font types, document styles and structure, word meanings, relationships between words and assumptions about text contents - expected contents, expected contents of certain parts or knowledge on relationship between text and field of its application.

(2) Intelligent interpretation of a drawing

Drawings, especially technical, could be scanned and recognized, much better and more precise, by using certain algorithms for determining location of textual parts of a drawing and its separation, or methods for analysis of scanned drawings in order of acquiring regular shapes, instead of set of lines, irregular in their shape, size and thickness, algorithms for recognizing fill patterns and similar [2] [3].

5. Comment instead of conclusion

By some available statistics (from year 1992) [5], it is estimated that there is over 15 billion of paper drawings used in different companies, which have to be used and controlled, and that only 13% of them are in electronic form. It is also estimated, that over 10% of those drawings are lost or misplaced, because of inefficient organization, and that, only in USA, about 43 million man/hours are spent on storage, search, copying and other manipulation of paper graphics documentation and space of over 1.5 million of square meters is used for drawings storage and saving.

There is a lot of legal and practical reasons to store drawings for several years, including potential need for drawing editing. Drawings used in machine construction, had to be treasured as long as machines are produced, and even later, because of maintenance. The same, but for much longer period, stands for architecture drawings or civil engineering, for example.

Introduction of CAD systems, aimed for improvements in a field of productivity in drawings creation, easier usage of graphics libraries, easier storage, editing and communication with drawings. But, need for communication with companies not using electronic systems for picture manipulation, forced a situation in which every company had to keep people, offices and working methods, for handling both paper and electronic drawings. Consequently, instead of increase in productivity, that usually lead to duplicated capacities and decreasing of efficiency, because of a need for cooperation between two very incompatible, parallel systems.

Everything mentioned, clearly shows urgent need for creation and usage of efficient system for storage, manipulation and control of different graphics formats, toward which this paper hopefully leads.

References

[1] A. Dengel, *Stepping from Automatic Spelling towards Automatic Reading*, International Summer School "Information Technologies and Programming" Sofia (1992).

[2] S.H. Joseph, T.P. Pridmore, *Knowledge-Directed Interpretation of Mechanical Engineering Drawings*, IEEE Trans. on Pattern Analysis and Machine Intelligence **14, No. 9** (1992).

[3] R. Kasturi, S. Bow, W. El-Masri, J. Shah, J. Gattiker, U. Mokate, *A System for Interpretation of Line Drawings*, IEEE Trans. on Pattern Analysis and Machine Intelligence **12, No. 10** (1990).

[4] K. Kukich, *Techniques for Automatic Correcting Words in Text*, ACM Computing Surveys **24, No. 2** (1992).

[5] T. Maxwell, *Engineering Drawing Management*, DECSYM'92 - Latest Trends in Computing.

[6] J. McKendree, J.M. Carroll, *Proceedings of CHI'86 Human Factors in Computing System*, ACM, 1986.

[7] Z. Putnik, *Intelligent HELP System as a Help in Educational Process*, Proceedings of IV Symposium "Informatics in Education and new Educational Technologies", Novi Sad, (in Serbian) (1994), 86-91.

[8] R. Radiev, V. Dimitrov, N. Marinov, *A System for Development of Intelligent Interfaces*, Proceedings of 17 School with Conference Information Technologies and Programming, Sofie, Bulgaria (1992).

Zoran Putnik, University of Novi Sad, Faculty of Science, Institute for Mathematics, Trg D. Obradovića 4, 21000 Novi Sad, Yugoslavia

FILOMAT (Niš) 9:2 (1995), 367–376
Filomat '94, Niš, October 22-24, 1994
Geometry. Computer Sciences

ONE METHOD OF IMPLEMENTATION OF LISP INTERPRETER TO TRANSPUTERS

Jozef Kratica

ABSTRACT. *The paper describes one method of implementation of LISP interpreter to transputers. Developed interpreter contains standard functions common for almost all LISP versions. Architecture is binary tree message passing. Implementation was developed on transputer parallel C language (ANSI C with procedures for interprocessor communications and synchronization). Part intended for evaluation of functions (expressions) was parallelized, but I/O operation and parsing were sequential. This is caused by the technical limitations of transputer systems, because I/O operations can executed only by first transputer, and interprocessor communication is slow. Maxima increase in speed equals 6.5 times, on transputer system with 17 transputers T800, by as compared to single transputer T800. That increase in speed is obtained for recursive problems demanding much computing. Small increase in speed is obtained for problems with more I/O operations.*

1. Implementation method

In LISP implementation on uniprocessor machines ([2], [3]), the basic part for parallelizing is part for evaluating expressions (functions). Provided that only first transputer can perform I/O operations, these operations (I/O) must be executed sequentially. Parsing functions are also executed on the first transputer, because interprocessor communication is slow. First transputer sends function definitions to other transputers when they need them (when other transputers evaluate functions).

Technical limitations of transputer systems are ([7]):

a) Every transputer have 4 links to other transputers;

b) Every transputer must be reset (one of its 4 links) by other transputer. Only first transputer is reset by the host.

c) Every transputer can reset maximally another 2 transputers, one by system, and the other by subsystem reset link.