stichting
mathematisch
centrum

MC

AFDELING INFORMATICA
(DEPARTMENT OF COMPUTER SCIENCE)

IW 166/81 APRIL

P.J.W. TEN HAGEN

COMMUNICATING GRAPHICS

Preprint

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics subject classification: 68B99

ACM-Computing Reviews-category: 8.2

Communicating Graphics *)

by

P.J.W. ten Hagen

ABSTRACT

Communication of computer graphics information via public data networks is analyzed. Communication levels are derived from functional levels in existing graphics systems for viewing. The relationship between communication of graphical data and standards for graphics programming are outlined.

KEY WORDS & PHRASES: computer graphics, communication protocols, graphics standards

---

*) This report is not for review; it has been published in Infotech State of the Art Report series 8 number 5.

1. Introduction.

By the end of 1980 my institute will be hosted in a new building. The new data communication equipment available there will provide me with a high speed (2400 Baud, or three lines of text per second) terminal connection to at least eight different operating systems, four of which are outdoors. At the same time my usual 300 Baud telephone line connections will become 1200 Baud connections. The new situation allows me to connect my graphics equipment to a wide range of graphics systems and application programs.

However, it is very unlikely that I will become a user of all these possibilities. The simple reason is that there are too many barriers left. Each of these systems has its own way of doing graphics. The obstacles I would have to overcome are :

- A pile of manuals, for each system.
- A multitude of communication protocols.
- Adaptation of the graphics satellite software to each system.
- A 2400 Baud connection which is still very slow for graphics.

Thus, in itself this development is not very exiting. However, it is only a small step in a very rapidly developing field (data communication). Bigger steps forward are bound to be taken so soon that there is reason to pay serious attention to the problems just mentioned.

The sudden increase in communications options is right now taking place in almost all European countries. It also brings similar problems to all these communities.

Every user is confronted with a large variety of facilities he cannot adequately deal with, because there is to much difference. The main problem is to find a uniform way of communicating with this large variety of systems. In order to solve this problem there must be developed a methodology for communication that can be used as a guideline. This is especially true for that most promising form of communication called interaction, which means communicating with a computer program in real time.

Allthough in the end capacity of data transmission will not be a serious problem, in the past it has been the limiting factor. On short term (next few years), it will certainly remain a factor which will have serious influence on every system for communication. In this lecture we will outline what steps have been made on the long road to uniform communication systems. They will be evaluated with respect to transmission speed and the new ways of presenting graphical data.

First however the concept of communication will be explained.

2.0 What is communication?

Taking the user view of a computer system as a starting point, the communication component of a system gives the user acces to all the tasks and functions of the system. The acces is provided by:

- the command language, and
- the dialogue.

The command language defines the symbols, in the widest sense of the word, which are used to denote the concepts of the task level. The dialogue describes which physical device manipulations by user and system are associated with each of the symbols of the command language.

At the command language level the representations of the concepts to be communicated are given form and structure. The concepts are associated with tasks and functions (in user terms!) to be performed by the system. Form and structure are to a great extend determined by this association. It is here that the decision is made how (in what "chunk"s, absolutely or incrementally, etc) concepts are embedded in the command language. The dialogue specification is a mapping of all representations onto executable sequences of actions and reactions. The dynamic behaviour of the system is defined by the (interactive) dialogue language. It is through the dialogue that physical aspects like transmission speed, coding, device characteristics etc. are taken into account.

The two components of communication (command and dialogue) are not always in agreement with each other. Very often the command language has to be adapted (degraded) in order to produce an acceptable dialogue on the available equipment. Until quite recently the two components have been studied separately. One reason is that command languages as such have received little attention.

So far communication has been discussed in general terms, not specifically graphical. Computer graphics however, has established itself as the most important means of communicating with computer systems. In the sequel we will discuss the two aspects of communication in the graphics area. On the dialogue level the main problem will turn out to be transmission speed, on the command level progress is most of all hindered by lack of methodology.

3. Aspects of Graphics Communication.

A graphics facility is a means for providing a user of an interactive system with a language in which his problem can be expressed directly (e.g. molecules and compounds for a chemist, streets and towns for a cartographer etc.). Graphics provides a realistic representation of the user concepts. In the course of a conversation these representations are transmitted between user and system.

In the sequel it will be assumed that the graphics equipment and the system are physically remote. Communication therefore implies data transmission. This is a very realistic assumption. Processing graphical data and application programs which use graphics usually require a large computing capacity. On the other hand each user wants to have his own configuration, tailored to his needs. The very rapidly growing community of graphics users will make this assumption become more and more realistic.

One of the characteristics of the recent past is the fact that data transmission has been a severe bottleneck. Nevertheless there have been other problems which have overshadowed the need for more transmission capacity. These problems are related to the high costs of software development on the one hand and large variety of graphics equipment on the other. A lot of effort has been put into graphics systems which are:

- device independent, which means that such systems can serve most graphics devices;
- portable, so they can run on many installations;
- general purpose, so that many applications can use them.

Each of these systems can be characterised according to the level of communication they support. In order to explain the term level of communication, the command language is (as with any language) thought to consist of a lexical, syntactical and semantical level. The lexical symbols of a language are the smallest units that can be identified. A human produces them by reflex, i.e. unconsciously. As far as graphics equipment is concerned they correspond to the basic operations by which they are controlled or operated by user or system. Examples are for input: keyboard strokes, pointings with a light pen, for output: drawing an elementary vector, a character or a pixel, blinking an item, moving pen or beam, etc..

3.1 Transmitting Lexemes

The most primitive physical realisation of communication is to control a graphics facility directly from the application program. For this only a code of the basic graphics operations has to be agreed upon. The device driver is so to speak resident in the host computer. This situation is comparable to a full duplex connection of alpha numeric terminals.

It is illustrative to compare the speeds required for graphical and alpha numeric displays.

An alpha numeric display is usable with a transmission speed of 110 bits/s. Smooth operation requires speeds in the order of 300 - 1200 bits/s. The need for the higher speeds becomes more urgent when the display has a local storage capacity for several pages. This encourages a user to list and send whole pages.

Suppose a coordinate value on a graphics terminal screen requires 4 bytes (2 for x, 2 for y). To draw an elementary vector would require 5 bytes (one extra to specify the function). A 110 Baud connection allows transmission of two vectors/s. A 1200 Baud connection 20 vectors/s. A very simple picture like an annotated graph would then require a time varying from tens of minutes to minutes respectively. Complicated pictures like maps and engineering drawings would take hours rather than minutes. Input activities like sampling from a digitising tablet or light pen tracking would be impossible, because real time demands cannot be met.

Nevertheless there is a considerable number of people which operate a graphics terminal under such circumstances. It is obvious from this that people indeed like pictures.

Speeds up to 1200 Baud are in most countries the maximum available on public data nets in the near future. In the sequel we will see how a more effective use of such a connection can be made by increasing local intelligence. However, to complete the comparison the promising new range of raster graphic displays will be analysed, when it is used under similar conditions(i.e. communication is at the lexical level).

A raster display contains an addressable grid of typically 1000 points square. Each point (called pixel) can have colour and intensity which can be chosen from a table of say, 128 different values. Therefore a point value takes one byte. A full screen requires transmission of 1000 kbytes. On the lowest level a screen is "painted" row after row. Hence, to change a picture a rewrite of the whole screen may be required. In the 110 - 1200 Baud speed range this would take from 24 to 2 hours. By using a clever coding scheme, less resolution etc. this could be reduced by a factor 3 - 10. At best pictures could be produced in ten minutes. The conclusion is that for raster displays transmission speeds in the order of 9600 Bd are a minimal requirement.

The overall conclusion is that the quality of communication at this level is completely determined by transmission speed.

3.2 Communication at the Syntactic Level

Increasing local intelligence means that the graphics terminal is capable of decomposing output primitives in elementary actions such as generating display code, plot steps, or scan conversion. In a similar way on input side a sequence of actions leading to say, locating something on the screen may be replaced by a report of the final location.

Although the principle reasons for increasing the level of communication (as given earlier) are not to improve the efficiency of data transmission, in practice this effect can be obtained by taking advantage of the situation. For instance, conversion from lexical to syntactical units leads to more compact representations as well as greater tolerances in response time.

Typically the response times for echoing at the lexical level are in the order of 50 ms whereas syntactic actions allow times from 1 to 4 seconds. There is a strong correspondence between response times and the speed with which the user normally produces these units. Natural responsiveness of the system allows a user to concentrate on semantics, being unaware of lexics and syntax most of the time.

3.2.1 Meta Files

A metafile contains graphical data organised according to a well defined device independent specification. A metafile can be stored, written on magtape or transmitted via communication links.

There exist a number of recently published metafile definitions. The two most important metafile proposals, which combine facilities present in more ancient ones are contained in the two proposals for a graphics standard. These standard proposals are GSPC79 CORE proposal and DIN GKS proposal. Both have been published very recently (august, september 79). In section 4 some more attention will be paid to both standard proposals.

The GKS meta file (the GSPC meta file looks very much the same) is a good example of syntactic units for graphical conversation. The metafile which is output oriented only (reason why it is often called plot file), has the following primitives:

- polygon: A sequence of connected line segments, defined by its vertices.
- polymarker: A sequence of (the same) marks placed at the points specified.
- string: A sequence of strings starting at the point preceding each string.
- fill polygon: A shaded area as defined by the closed polygon.
- pixel array: A rectangular grid with a colour specification for each individual cell of the grid.

By grouping primitives in reusable segments a further compactification can be obtained. Reuseable segments are stored "locally" at the graphics terminal. In a similar way descriptions for pen representation containing colour, line width, line type intensity, etc. as well as text representation containing font, quality, size, spacing, etc. are grouped and stored for multiple invocation. Moreover, on reuse of a segment linear transformations may be applied.

Together these facilities lead to a considerable more compact representation of computer generated pictures.

3.2.2 Input Units

For input the standard contains syntactic primitives with a different organisation. The means for compactification are less powerful. There are two reasons for this:

- a human operator does not(with todays equipment) produce data at high rates;
- the application program may assist a user in producing pictures he wants to discuss. In doing so the output stream is used to visualize user defined pictures.

Like for all alpha numeric input, user input actions are monitored and processed at a relatively low level. This situation may well change when image processing techniques are going to be mixed with graphics at a remote terminal.

The input units present in GKS are based on the concept of a logical device. A logical device is a virtual device defined for a particular purpose. It represents all physical devices with the same purpose. There are five logical devices in GKS. Below they are listed with the corresponding primitive input unit they deliver:

Logical device	Primitive unit
Locator	Array of positions (x-y pairs)
Valuator	Array of scalars
Choice	Array of (push) button numbers
Pick	Array of pick identifiers
Text	Sequence of strings

These primitives illustrate that input remains below the picture level. Only a locator sequence or a pick (= name of a subpicture on the screen, selected through some form of pointing) have a direct relation to graphical data, but they do not directly represent visual objects.

The locator device is capable of inputing data generated by digitising. Compactification can be obtained by applying data reduction methods on the sampled data and generate the array of positions as a result. In other words, the standard can handle fast input devices if sufficient local intelligence is provided.

4 Interaction and data transmission.

In this section interaction and data transmission will be discussed. Basic facilities for interaction as present in current standard proposals will be analysed. These facilities constitute a fair characterisation of the state of the art. In the course of developing the standard these facilities have received much attention. This has resulted in a rather advanced basic package which, for a standard, is quite unusual.

The examples will be taken from GKS as well as GSPC-CORE. Both national standards proposals are based on the same principles. Both groups also have established a cooperation which very likely will result in merging the two proposals somehow into one international standard.

4.1 Dynamic Aspects of Output

For graphical interaction gradual picture change is at least as important as generating whole new pictures. The possibilities for writing a new picture can be deduced from section 3. Hence now gradual picture change will be discussed.

The unit of modification of pictures on the screen is the segment. A segment is a named sequence of output primitives. Individual segments can be dynamically changed through the following functions:

- deletion
- making a segment visible or invisible
- renaming
- insertion of a copy into another segment
- transformation (e.g. scale translate and rotate)
- blinking or unblinking
- making them to cover or be covered by other segments (raster displays)
- last but not least making them selectable for pick.

If a segment is stored locally and these functions for manipulation are implemented as part of the local intelligence then these facilities are very well usable even with low speed (< 1200 Bd) transmission. Only the commands would have to be transmitted. If they are not locally implemented, in most cases retransmission of the segment or the whole picture(!) would be necessary. For storage tube displays the need for local facilities is even more pressing because every change of a segment implies rewriting the screen, either directly or delayed. The latter means that updates can be batched. The standard contains this facility especially for saving transmissions.

4.2 Dynamics of Input

The dynamic aspects of input are restricted to echoing and control by the system over the aspect of initiative.

Initiative is specified by two modes of input. Input can be event driven which means that the input primitive is put into the input queue as soon as it is produced by the terminal operator. Putting something in the input queue may cause an interrupt or a program wake up etc.. Input can also be sampled which means that the system may decide at any moment in time to read from an input device, irrespective of what the operator is doing. Data transmission capacity has a strong influence here on real-time behaviour:

- there is a delay between sample command and actual sampling
- there is a delay between producing an input and the subsequent event report. As far as the standard is concerned real-time aspects are not specified because they are totally application and implementation dependent.

For echoing there are two possibilities. On the lexical level echoing is to be provided locally. It can be influenced by the program in two ways:

- select echo (including none)
- select area on the screen for echoing.

Echoing on the syntactical level is thought to be application dependent and therefore not specified.

In conclusion one might say that within the standard echoing can be provided by lexical half-duplex mode which does not require any data transmission (again if sufficient local intelligence is provided).

5 Other Media.

From the point of view of data transmission two other types of pictorial information exist, namely, facsimile transmission and video.

The first can be characterised according to section 3 as communication at the lexical level of scanned information. As we have seen, for the near future the capacity of transmission will not be sufficient to incorporate this form into graphical data processing. Hence, the expectation is that facsimile transmission will remain a special purpose system for picture data transmission in areas where it has been proven useful (news, signature verification, etc.).

Video graphics will through its compatibility with raster graphics have great opportunities. In fact they work so happily together that each one will strengthen the position of the other.

Video can be mixed with raster graphics in two ways:

- on the screen by superposition

- in scan memory, so that video can be manipulated frame by frame.

The first way brings new possibilities for digitising, data extraction and annotation.The second has big opportunities in for instance animation.

New digital storage techniques (video disc) allow for storage of graphics and video on the same medium.

The reaction of the information processing world to all this is not yet known since it is just marching out of the laboratory. However, certain that new software systems yet have to be developed to cope with all these facilities.

The fact that the equipment very likely is going to be mass produced, asks for enormous increase in data transmission capacity (say, video to replace telephone). Thus for a longuer future datatransmission is going to remain a sparse resource. One therefore may expect the trend towards further distribution of processing power(with is another word for local intelligence) to continue.

6. Text.

Teletext is a one way communication of comparatively small amounts of information for mass consumption. Something like transmitting newspaper headlines. It is not attractive because of its quality but because it is so cheap. In a few words: I don´t like it. It is not integrated in other forms of communication, it is a parasite of television.

Viewdata is the right solution for the future. Its relation with graphics is a very interesting one. First of all raster techniques allow for mixing of both. Secondly graphics will have a sound effect on viewdata in two ways:

- Information can be presented in parallel, this becomes obvious when graphics enters the scene.

- By using lay out and other 2D facilities the presentation can be enhanced with three important aspects of communication:

  -- anticipating the next question

  -- giving (much) more information than was asked for

  -- present cues for reiteration.

It is believed that this effect of graphics on viewdata can be so strong that graphics may catch up with viewdata and become the most important portion.

As a special case one may imagine the possibility to replace alphanumeric terminals with low cost raster displays. Such terminals will (given adequate facilities like editors, data representation and file handling) increase programmer and/or operator productivity to a great extend.

7 Semantic Conversation

Semantic conversation is not within reach of the current or near future state of the art. Interaction and communication at the syntactical level are not yet well understood. This looks similar to the period of assembly language programming which hindered understanding programming. May be the simplifications made possible by the new developments will help finding a methodology of man machine communication.