

AUTO-HERALD

A Program for the Construction of Heraldic Drawings

By C.B.Bayliss M.Sc

University of Birmingham Centre for Computing
and Computer Science

Abstract

In this paper I shall describe the main features of a program which constructs heraldic drawings. A description of a coat of arms is entered by a user in conventional heraldic terminology. The coat of arms is then constructed from a library of stored objects and fields and drawn on a terminal.

Facilities exist for updating the library, and a picture editor is provided to allow a user to define new objects and fields.

Contents

Introduction
Describing a coat of arms
Method of coat of arms construction
Picture editor
Library
Device dependence and program structure..
Conclusion

Illustrations

Fig 1 - Diagrams showing method
of coat of arms constr-
uction
Fig 2 - Overall structure of the
program

Introduction

In heraldry there are many different coats of arms which use a variety of objects in different combinations and colours. A coat of arms consists of a field (the background), which can be a colour, metal, fur or pattern which may have one or more objects placed upon it. Such objects are referred to as "charges" in heraldic terminology. A field may be divided, for example by quartering it or halving it. Each part of the field may be a different colour or pattern. Coats of arms are normally displayed on a shield.

Colouring used on a coat of arms is called a tincture. There are five commonly used colours:- vert (green) azure (blue), gules (red), sable (black) and purpure (purple). Two metals are used:- or (gold), often represented by yellow, and argent (silver), often represented by white. Pigments are sometimes used and the main ones are tenne (orange-brown), sanguine (blood-red) and murrey (violet). Furs can be used on fields and these are often represented by a plain colour with a pattern of tails in another colour (eg. white with black tails for ermine).

Because of the great variety of coats of arms, a method of forming a concise but accurate description became necessary. Conventions have developed for forming descriptions using terms from medieval English and French, although modern terms have crept in later. Such descriptions are called blazons. The ordering of the parts of a blazon is not strictly defined, conciseness sometimes being considered of greater importance.

The program was developed to accept a blazon from a user, to construct the coat of arms from a library of stored charges and fields and to display it on a suitable terminal. A colour raster display terminal was chosen as the main output device, with hard copies obtainable from character printing terminals using different characters for different colours.

I have chosen to call the program "AUTO-HERALD" in the absence of a better name and because of a strong dislike of acronyms.

AUTO-HERALD is a fully interactive program and has been designed so that it can be used by people with little or no experience of computers or heraldry. Helpful features are available, such as relevant

information for the section being used on typing "HELP", and clear error messages. Help not needed is not given so that users with more experience do not find the program annoying.

Describing a Coat of Arms

A major problem was deciding how to convert a blazon into instructions for a program to construct a coat of arms.

A direct attempt to translate a block of text was rejected for the following reasons. Firstly, a blazon is basically a block of English text without a clearly definable syntax, and a program to cope with comprehension of English even in a limited sphere was far beyond the available time and resources. Secondly, if the syntax and format were to be restricted sufficiently to make this approach feasible it would be necessary for a user to learn in detail exactly what is and is not acceptable. Thirdly, I believe that most users would not wish to input descriptions in this way as few would be proficient at constructing blazons. If a user were to make an error this might not be detectable until the whole blazon had been typed.

The approach followed was to use a particular ordering of the parts of a blazon to form the basis for a series of prompts to interrogate the user. The field is described first, then charges and their tinctures. In the case of a divided field, the type of field is given first, followed by a description of each part of the field in turn as if it were a field in itself. For example, if a user wants to describe a coat of arms consisting of a red rampant lion on a gold background, the interaction is as follows:-

```
Field >OR
Charge >LION RAMPANT
Tincture >GULES
```

The user input is shown in upper case to the right of the prompt ">". A traditional blazon for this coat of arms would be "Or, a lion rampant gules".

A better example can be seen from construction of the University of Birmingham's coat of arms. This needs the output that is produced from the following description.

```
Field >TIERCED IN PAIRLE REVERSED
DEXTER                >GULES
Charge >LION RAMPANT DOUBLE-HEADED
Tincture >OR
SINISTER              >AZURE
Charge >MERMAID WITH COMB AND GLASS
Tincture >OR
IN BASE                >SABLE
Charge >BOOK OPEN AND BOUND
Tincture >OR
```

The upper case words again form the blazon. Those to the left of the prompts are supplied by the program, and are obtained from the header information in the file containing the appropriate field description. Names for the charges may vary, so facilities exist for describing synonyms, and existing entries can easily be changed. Abbreviations (minimum ten letters) for charges are accepted if they are unique in the letters given and terminated with an asterisk. Standard heraldic abbreviations for tinctures are supported, as are modern English terms.

Coats of arms can be of arbitrary complexity. It was necessary to place limits on the complexity of coats of arms produced by the program.

Auto-herald can produce the following coats of arms:-

1) A plain colour, metal or fur with or without a charge placed upon it.

2) A divided field comprising more than one area of colour, metal or fur which may have a charge on each area scaled to the appropriate size. One full-sized charge may take the place of others, or charges may be totally omitted.

Method of Coat of Arms Construction

The user is first interrogated for a field description. If the field description is not simply a colour or metal, the library is searched for furs, then for divided fields. If the search is unsuccessful, the user is given an appropriate error message.

If the field is a simple colour or metal, or a fur, the user is asked for the charge and then its tincture. The charge is then loaded into a buffer area, its colour being changed appropriately, and stored in a second buffer. (Buffers used to store pictures consist of arrays of integers and are dimensioned to 256X256. Each integer represents the colour of a particular point on the picture stored.)

If the field is divided, the header information in the appropriate file is used to prompt the user for a description of the contents of each part of the field in turn. Following each of these descriptions, the charge is loaded in the same way as a single charge but is scaled and moved to the size and position stated in the field file before being moved to the second array.

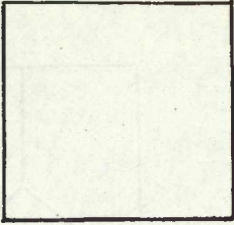
The field is then loaded into the second buffer area and the colours adjusted accordingly. Furs in the divided field are treated differently. At this stage, a large number is loaded into the buffer locations corresponding to the section containing the fur.

Once the field has been loaded, the contents of the second buffer are placed on top of the first, blanks being treated as transparent. A mask is placed over the picture, producing the shield shape. The buffer contents are then placed in the second array.

If any areas of fur exist, the appropriate fur pattern is now loaded into the main buffer and coloured. Locations corresponding to fur in the picture are then made equal to their counterparts in the main buffer.

The main buffer is then coloured grey before the coat of arms is returned to this area for output to a colour terminal. If the output is for a printer, the coat of arms is scaled down accordingly and the colours are converted to characters.

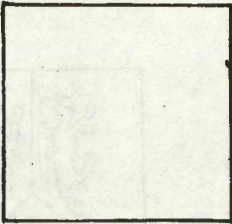
The procedure is probably easier to follow from the diagrams in Fig 1.



1



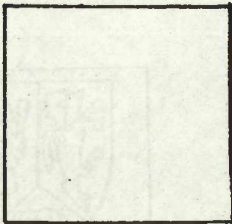
1) Double headed lion rampant is loaded into the picture buffer.



2



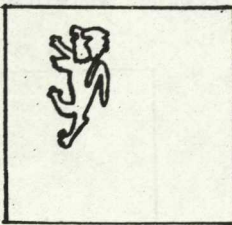
2) Contents of picture buffer are scaled.



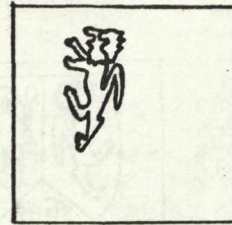
3



3) Scaled block is moved.



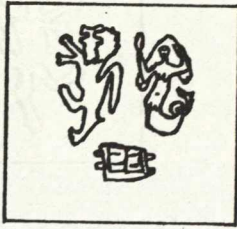
4



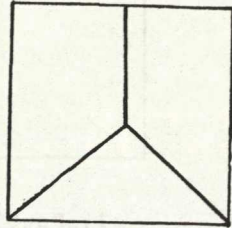
4) Picture is stored in secondary storage buffer. Picture buffer is then cleared.

Steps 1 to 4 are repeated for each charge.

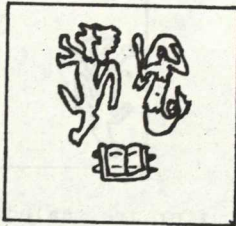
Fig 1



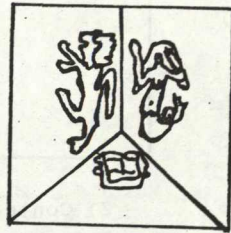
5



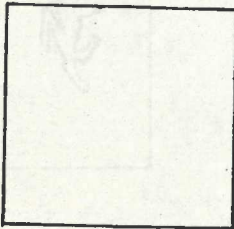
5) Field is loaded and colours changed.



6



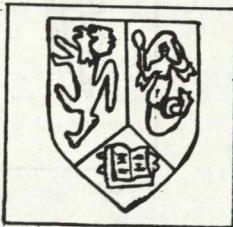
6) Contents of picture buffer are placed on top of field.



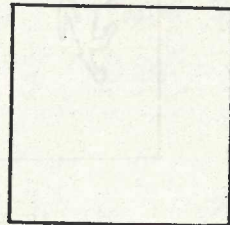
7



7) Mask is loaded to produce shield shape.



8



8) Complete picture buffer copied to secondary storage buffer. Furs are dealt with and picture buffer is filled with grey before copying back from storage buffer. Output to screen follows.

Picture Editor

A picture editor is supplied to allow the input of new charges, furs or fields and store them for later use. The picture editor was used to set up the initial library of heraldic devices.

Line drawing, rectangle, solid circle and triangle generation and general purpose fill-in routines are provided. The user has a keypad which is used to move the cursor and selects functions using single letter commands.

In addition to these basic drawing functions, the user can delete, move, copy and scale whole areas of picture and enquire about the cursor co-ordinates. Drawing aids such as a grid, shield and centre cross are provided which are displayed on the screen but not stored.

New charges are drawn full size in their correct position and consist of a colour to be defined at run-time by the heraldry routines and one or more permanent colours, which are useful for adding black lines to drawings.

Fur patterns are defined by drawing the pattern in two different non-permanent colours. Fields are defined by drawing areas in different colours, using the numerical order of the colours to match the order in which the areas are described.

Pictures drawn using the editor can be stored in files. The file consists of a horizontal line scan of the picture. Each colour is represented by an integer. The position to which the colour extends is recorded alongside this. A blank line in the file signifies the end of a scan line of the picture.

For example, a red line would be stored as follows:-

```
1 256
```

A blank space surrounded by red would appear as follows:-

```
1 120
0 140
1 256
```


Three identical lines together would be represented as :-

1 120
0 140
1 256

1 120
0 140
1 256

1 120
0 120
1 256

The limit of resolution of AUTO-HERALD is 256X256, being high enough to give reasonable pictures and low enough to keep processing time of instructions down to a reasonable amount for an interactive program.

Library

Once a new charge, fur or field has been created and stored, it must be entered in the library catalogue before it can be used by the coat of arms construction routines. The catalogue is an alphabetically ordered list of descriptions and their associated files.

Fur and field entries are appropriately marked to distinguish them. More than one description may map to the same file, allowing for synonyms. Colours are stored with fur descriptions in the catalogue so that one file may be used to form more than one fur.

On entering a field description to the catalogue the user is asked for names of areas within that field and the positions and scaling factors to be applied to charges placed within them. This information is added on extended records at the start of the file containing the field.

A user may use the library facilities to list parts of the catalogue in order to determine what is available in the stored library. Auto-herald will allow a user to define his own catalogue to be used in addition to the one provided.

Device Dependence and Program Structure

The program was written on a DEC-20 in FORTRAN, which allows for the possibility of implementation on a different system without major modifications. The colour terminal used for output is a Sigma GOC 525 raster display terminal. The program also runs on a Sigma T5680 terminal. Only a few primitive drawing instructions are used in easily replaced subroutines, so to add a new device to the repertoire of Auto-herald is not a major task.

The program has been written in a modular fashion, facilitating enhancements and improvements. The overall design is shown in Fig 2

Conclusion

AUTO-HERALD is a basic interactive system which constructs coats of arms. The program could be extended to allow for more complex coats of arms and even supporters, but the implications of increased processor time must be considered carefully in an interactive environment. A quartered coat of arms can take a few minutes to construct using the present version.

It can be seen that the coat of arms construction routines could be replaced with other routines without disruption to the rest of the program. This allows for the possibility of modification of the program in order to solve other problems involving the manipulation of similar graphical information.

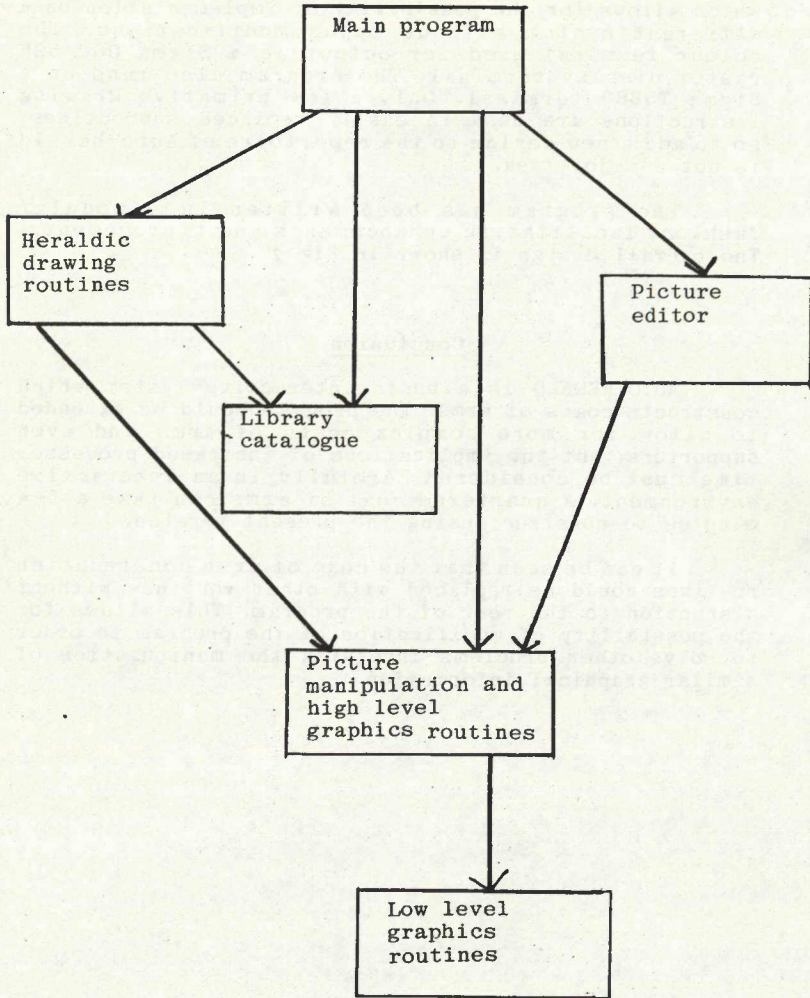


Fig 2 - Outline structure of program