Introducing Graphics Capabilities to Several High-Level Languages

NAM NG AND T. A. MARSLAND

Computer Centre, University of Hong Kong, Hong Kong
Department of Computing Science, University of Alberta, Edmonton T6G 2H1, Canada

SUMMARY

The development of basic support software for a graphics facility requires extensive programming effort, which is sometimes duplicated when the facility is made available from other languages. This paper explores an alternative, which has the advantages that only a single software package need be built for use by a variety of host languages, and no modification is necessary to their compilers. Fortran, PL/I and Algol W are the example hosts used in the feasibility studies, but the approach can be extended.

INTRODUCTION

The development of support for a graphics facility is complex and requires extensive programming effort. One obstacle to wider application of this area is the environment in which the facility is made available. For many applications, graphics operations are only one component in the total solution of a problem; invariably capabilities other than picture description and manipulation are required.[1] One can introduce graphics capabilities by inserting additional features into a host language,[2,4] or by providing special subroutines that can be called from the host.[5,6] Unfortunately, these approaches impose a serious restriction on users who do not have a good knowledge of the language from which the facility is accessible.

Ideally, one should be able to use graphics from whatever programming language is best suited to the application. For example, problems which involve complex data transformations may need a data-structure language. More commonly, the language is simply the one which the programmer uses most often. However, to provide each user with 'his' graphics language, the implementer of the support package faces the chore of reproducing the same basic software but tailored to each of several general- or special-purpose languages.

The above considerations have brought us to explore techniques for introducing graphics capabilities to high-level languages. Our idea is that statements from a single picture description language should be mixed freely with those of the hosts so that they appear as an extension. The advantages of this approach include:

(i) Only a single graphics facility has to be built for use among a number of high-level languages;
(ii) The picture description language can be simple and limited to graphics operations, leaving other capabilities such as data and control structure manipulations to the host;
(iii) Access to a graphics facility in a uniform way from several high-level languages allows a wider range of applications;

0038–6644/78/0508–0629$01.00

Received 19 October 1976
Revised 5 January 1978