The CGM Application Profile for CALS: Current Specification and Major Issues

L. R. Henderson

Abstract

The Computer Automated Logistics Support (CALS) is a DoD initiative to establish a near paperless logistic technical information system. The CGM has been designated as a standard interchange mechanism for 2D vector graphics. The necessity of a CGM Application Profile (AP) has been identified and an initial draft has been produced. As the profile has not yet been published, significant portions of it are presented here, along with discussion of major issues (resolved and still pending) in its formulation. A high degree of compatibility with the TOP Application Profile has been achieved. There is need to go further with functional extensions to meet CALS user requirements. Work is in progress on some functional extensions, which should be adopted into the profile when they are mature and stable enough.

1 Background

The Computer Aided Logistics Support (CALS) program is an initiative within the Department of Defense to solve the problems arising from its current paper-intensive logistics processes. The solutions involve automation of those processes, with the goal of producing a near paperless logistic technical information system. Integration of the many efforts already underway independently in various agencies is a primary requirement. The integrated solutions will have to deal with large investments of existing technology (e.g., CAD/CAM systems) and must make use of large existing databases (e.g., CAE and CAD design data for insertion into technical manuals).

CALS, through the National Bureau of Standards (NBS), has made a major effort to identify and incorporate national and international standards for data and application interfaces within CALS application environments. Graphics standards are one such

A. Mumford et al. (eds.), *CGM in the Real World*
© EUROGRAPHICS The European Association for Computer Graphics 1988

category of interface standard, and CGM in particular has been singled out as having an key role to play. It's major uses are anticipated to be for recording and interchanging technical illustrations, project planning charts, raster-to-vector converted scanned images, etc. for later review and incorporation into technical documentation, maintenance manuals, and management and administrative documents of all kinds.

2 Overview of the Profile

A high priority of the CALS profile was avoidance of proliferation of incompatible specifications. This meant in particular avoiding incompatibility with the TOP AP for CGM. This was accomplished in large part by collaborative effort between the experts of the two projects.

The TOP profile forms the basis for the CALS profile. The CALS profile corrects some problems (editorial errors, etc.) that still exist in the TOP profile. In addition, it adds some conformance requirements for generators and interpreters that do not exist in the TOP profile. In particular, two levels of conformance for interpreters are specified:

- preview-quality conformance - the fallback actions of Annex D may be used, monochrome may be substituted for color, etc.
- publication-quality conformance - the semantics of the Basic conformance set of the profile must be exactly rendered.

The TOP profile, and the initial version of the CALS profile as well, are still functionally lean when measured against the requirements to efficiently support the CALS constituency. As part of another CALS/NBS project (see the paper by Carson, "Extending CGM for CALS," in these proceedings), functional extensions to CGM have been recommended in a number of areas. The mechanism for achieving these is the Graphical Registration of ESCAPE and GDP elements. A number of proposals have already been submitted.

The initial Application Profile for CALS recommends that the profile should be amended or extended to include such functionality as soon as it is stable enough in the standards processing pipeline. Similarly, it is recommended that CALS begin adopting Extended Metafile (ISO Addendum 1 to CGM) functionality as it stabilizes (i.e., those functions, such as symbol library facility, that are important for CALS constituents).

The most serious functional deficiency of CGM, in an environment of automated publishing and technical illustration, is the lack of text fonts of decent quality. The draft CALS profile does not solve this directly (it basically follows TOP here, with some conformance guidelines); but it recommends following the work of ISO draft standard 9541 on font and character set specification as it stabilizes. In particular, the profile recommends: future work for CALS should focus on a method of specifying an adequate and useful set of fonts; and this should be done in a way which relieves conformers to the profile of financial obligations to font copyright holders.

3 Introduction

The following sections, defining application profiles and stating the need for same, are largely repeated from the CALS draft profile.

3.1 Overview of CGM

The Computer Graphics Metafile (CGM) standard, ANSI X3.122-1986 and ISO 8632/1-4, specifies the syntax and semantics of a standard file format for storing and communicating computer graphics pictures. By intentional choice of scope, it limits the specification to the syntax and semantics of a set of CGM "elements" for the device-independent description of computer graphics pictures.

In the year that it has been an ANSI standard CGM's use and its incorporation into other standard interface and exchange specifications has been increasing. There are over two dozen implementations existing or known to be in progress in the US alone (there are more internationally). It has, of course, been specified as the 2D vector graphics exchange mechanism for CALS. It has been designated as a Federal Information Processing Standard (FIPS). It has been incorporated as the graphical metafile of the MAP/TOP V3.0 specification. It is designated as the Geometric Graphic Content Architecture of the ISO compound document standard (ISO 8613, currently in DIS stage), known as "ODA/ODIF". It has been specified for vector graphics interchange by MIL-STD-1840.

3.2 The Need for CGM Application Profiles

Even though it is achieving widespread acceptance as a graphical standard there are still problems using CGM in some environments. The syntactic specification in the CGM standard is complete and unambiguous. It is, as well, redundant in the sense that there are three distinct encodings of the same functionality - binary, character, and clear text (according to CGM, each encoding is tailored and optimized to certain computing environments and applications area).

The semantic specification is less complete. The expected overall results of using the geometric primitive elements are well enough specified. However some of the finer details, such as the precise appearance of joints and endpoints in lines, are unspecified. This underspecification of semantics was intentional on the part of the designers the CGM - it allows a wider range of existing systems to be accommodated and makes the standard more adaptable. CGM was written for the needs and philosophies of a diverse clientele.

On the other hand, the semantic ambiguity does mean that there will be no single correct interpretation of a given CGM, and hence it will be difficult to unambiguously describe an intended picture using CGM. This is a distinct drawback in certain applications environments. The CGM application areas of Technical Illustration and Technical Publishing, which are central to the CALS effort, represent such environments where unambiguous semantics are critical.

There are further sources of uncertainty in using CGM in an application environment. The CGM standard specifically excludes standardization of the behavior of metafile generators and metafile interpreters. In doing so, a certain unpredictability of results is introduced into the graphics system viewed as a whole - for example, CGM generators serving GKS (Graphical Kernel System, ANSI X3.124-1985) clients in the product lines of two different vendors might map out-of-range attributes differently.

The CGM is a basically suitable protocol for application areas such as technical publishing and illustration (ISO 8613, ODA/ODIF adopted it for example). But these two sources of ambiguity in using CGM - incomplete semantics and non-specification of the behavior of generators and interpreters - do mean that some further specification (beyond that in the published standard) is required in order for the use of CGM to be effective, efficient, and unambiguous.

Such a specification is precisely what an Application Profile (AP) consists of.

In the case of CGM, an AP can:

1. specify complete semantics;
2. specify the behavior of CGM generators and CGM interpreters;
3. extend the functionality by ESCAPE or GDP elements.

An AP at least specifies minimal and maximal requirements for generators and interpreters, and ties down all implementation dependencies of the CGM. As the name suggests, an AP is a set of specifications appropriate to a given application environment.

One such CGM AP has already been targeted and printed - the CGM Application Profile of TOP (Technical Office Protocol) that is being endorsed by a number of major industrial constituents and has been incorporated into the MAP/TOP V3.0 specification.

For CGM to be used effectively in the CALS Technical Publishing, Administrative Publishing, and Technical Drawing applications, an AP has been designated for CALS as well.

4 Issues and Objectives in Formulating the Profile

4.1 Conceptual Issues in Specifying the CALS Application Profile

There are two categories of specification that were considered in specifying the CALS Application Profile of CGM:

1. resolution of ambiguities in the metafile and in the behavior of generators and interpreters;
2. extension of the CGM functionality to handle perceived functional deficiencies in the standard.

There are two ways to accomplish the second task: firstly, by definition of ESCAPE and GDP elements, which are then submitted for Graphical Registration; secondly, by revision of the CGM itself via normal processes (e.g., ISO Addendum 1, CGEM).

Any Application Profile must accomplish the first task. The second task is important for CALS constituents as well. The CGM is functionally lean when measured against the requirements of automated publishing and technical illustration. Almost any picture from these application areas can be represented. For example lines and areas can be used to represent in the CGM many of the "higher-level" entities of IGES. But the consequence of simulating the entities with very primitive geometric elements is a loss of efficiency and data compaction in the CGM.

In order to be an efficient picture mechanism in the CALS environment, extension of the functional capabilities of CGM is required. Such extension is taking place formally now within ISO (the Extended Metafile - Addendum 1 to CGM). Unfortunately, even the "fast-track" ISO addendum process is a slow process.

The CALS project has completed a study (see Carson's paper) to:

1. identify needed extensions to CGM, and;

2. prepare and submit Graphical Registration proposals for same.

Adopting such proposals into a profile at this time presents a dilemma. Doing so would surely result in their immediate use. While they are needed functionality, the proposals will be examined and likely modified by graphics standards bodies before they have official standing in the Standards arena. Implementations which use the proposals too early in the registration process will likely be non-standard when the proposals eventually complete standards processing.

The initial Application Profile for CALS therefore does not include any extended functionality, with a minor exception: the published TOP profile contains two specified ESCAPES, one of which is an encoding of a function that is stable in ISO CGI and in the CGEM. These have been adopted by CALS as well.

The TOP committee, working with CALS and ASC X3H3, removed from its report (among other things) two user-definable attributes, line and hatch style; this was done allow graphics experts to formulate the proposals and allow opportunity for adequate review. These are not included in the current CALS profile, but two specific proposals are attached to the draft for insertion into the Graphical Registration process.

In summary, the CALS CGM Application profile:

1. specifies semantics and syntax that are ambiguous or unspecified in CGM;

2. does not, except as noted above, specify extended functionality for CGM.

With regard to point number 2: the problem at this time is the immaturity of the proposals and the immediate need for an initial AP for the CALS community. The CALS AP recommends that extended functionalities should be included in a "Version 2" upgrade of the CALS AP in the near future - as soon as the work of the ISO and Graphical Registration committees has progressed far enough.

4.2 Relationship to the TOP Profile

Proliferation of "dialects" of CGM is clearly undesirable and contrary to the interests of the graphics industry and users. Accordingly, one principle priority in formulating the CALS profile was to realize an AP that is either identical to, or is backwardly compatible with, the TOP AP. In other words, where the APs overlap they should be identical, but CALS may be somewhat richer or may go further in specifying constraints.

Fortunately there is significant common interest and shared requirements between the sectors of industry represented on the MAP/TOP committees and the clientele of the CALS initiative, particularly in the areas of technical illustration and compound document exchange. This made collaboration and convergence of the profiles achievable.

4.3 Specific Goals of the Application Profile

Other specific objectives in the specification of the CALS CGM AP included:

1. A CALS metafile must be a legal CGM - CALS syntax must be a subset of CGM syntax and CALS semantics must be legal CGM semantics. This means, for example, that the CALS environment cannot assume or specify implicit element defaults that differ from the CGM standard.

2. The picture specified by a CALS metafile must be unambiguous. This means, for example, that private values of attributes (such as private linetypes) cannot be allowed, and private elements (escapes and GDPs, for example) must be prohibited.

3. The behavior of generators in producing a CALS metafile should be specified so that identical sequences of activity at the application level result in identical metafile contents (intermediate layers in a graphics environment may complicate this).

4. The behavior of interpreters in parsing and rendering CALS metafile should be as unambiguous as possible. This means that such things as fallback actions when the interpreter lacks capability, or fallback actions in the face of geometric degeneracies, should be specified.

5. The format ambiguities of the CGM, such as the "record size" of the binary encoding (unspecified in CGM) should be specified.

6. The CALS CGM AP should be rich enough to accomplish useful things economically.

7. The AP should be formulated with awareness of the evolution of graphics standards. In particular, the content of the Extended Metafile (ISO 8632 Addendum 1 - the first set of extensions to CGM, currently near DP stage) should be carefully followed. No specifications should be made in the CALS AP which compromise compatibility with these standard activities.

8. Similarly the activities of the Graphical Registration process should be tracked. Future compatibility must again be protected.

9. A CALS metafile should be self-identifying as such.

In some cases, these criteria are mutually contradictory - it is not necessarily possible to satisfy all of them at once.

5 Definition of the Profile

5.1 Basic Definition

The MAP/TOP V3.0 specification (references below are to the version most current on 1 August 1987) is the basis of the CALS Application Profile for CGM.

The CALS Application Profile for CGM comprises:

1. all of the specifications of the MAP/TOP V3.0 Application Profile for CGM with only modifications as specified in the next section;

2. extensions and additions as specified in subsequent sections.

5.2 CALS Modifications to the TOP AP

This section details those modifications to the baseline MAP/TOP V3.0 CGM AP that were required for CALS. Most of these are believed to be errors in the TOP profile, and are being submitted to TOP for erratum processing.

1. The restrictions of section 6.2.5 - command headers and text strings must use long form - are removed. (TOP experts indicate agreement with removing the restriction from the TOP profile as well.)

2. In section 6.2.6.2, Note 1, replace last sentence with; "If the metafile is a CALS Basic metafile, the substring CALS/BASIC-1 shall appear in the METAFILE DESCRIPTION element."

3. Section 6.2.6.2, table 6.2-2 - the CALS profile allows both floating point and the simpler fixed point reals in the Basic values of REAL PRECISION; the omission of fixed point from the TOP profile is believed to be a mistake.

4. Section 6.2.6.2, table 6.2-2 - typing error, "Font Index" should read "Font List".

5. Section 6.2.6.4, table 6.2-4 - the CALS profile allows both floating point and the simpler fixed point reals in the Basic values of VDC REAL PRECISION; the omission of fixed point from the TOP profile is believed to be a mistake.

6. Section 6.2.8.2 - the sentence should be replaced by: "The following Escape elements are required to be supported by all TOP Basic conforming CGM implementations." Clearly it is not intended, as literally stated, that all occurrences of metafiles contain the elements.

7. Section 6.2.8.2.1 - the last sentence should be replaced by: "The implicit default viewport is the largest area of the available view surface that has the same aspect ratio as the implicit default VDC EXTENT. The latter is square, so the implicit default viewport is the largest square area of the available view surface". ANSI/ISO CGI and CGEM, from which this element is taken, must be followed here. In those specifications the default window/viewport mapping is isotropic.

8. Section 6.2.10 - the second sentence appears to be an error, and in any case is an inappropriate sort of specification. Replace it with "If there is an element which the interpreter cannot understand or render, then processing should continue if possible with the next element following the problem element."

9. Section 6.2.9.2.1 - the specification of the data structure support for COLOR TABLE is incorrect and contradicts table 6.2-5, which is correct. The COLOR TABLE line should be replaced by "256 for COLOR TABLE (i.e., entries 0-255)."

10. Section 6.2.8.1 - as specified in the TOP profile, the only fonts available for predictable interchange are the Hershey fonts. These are of poor quality for modern illustration and publishing needs, particularly when the characters become large. The CALS profile considers any rendering of a requested font conforming if the rendering is "metrically identical" to the font metrics of the requested font. This means that the placement and alignment of the string and the placement, size, and shape of individual characters (i.e., the drawn portions of the character cells) are measurable identical. This would allow a good quality filled font to be substituted for a stroked Hershey font, for example.

5.3 CALS Extensions to CGM that are in the TOP Profile

TOP AP contains the following two extensions to CGM functionality:

1. DISABLE CLEARING OF VIEW SURFACE, section 6.2.8.2.1: this element gives control over an aspect of CGM that is unspecified - whether or not the view surface should be cleared upon the occurrence of BEGIN PICTURE BODY;

2. DEVICE VIEWPORT, section 6.2.8.2.2: this element gives control over placement of the virtual image on the view surface, and together with 6.2.8.2.1 allows production of composite images (several CGM pictures); its formulation is taken from the stable elements of the ISO CGI and CGEM work.

These elements are included in the CALS profile. This is contrary to the general policy of the CALS profile not to extend CGM functionality, but it is done because the extension has already been made in TOP.

A number of functional extensions were removed from early drafts of the TOP profile. Some were inappropriate or not of broad enough interest. Some were useful but needed to be studied by a wider audience and reformulated. Two were considered particularly useful. These are user-defined linetype and user-defined hatch style. User-defined linetype has been recommended for registration by CALS (see Carson report).

The initial CALS profile recommends that these two items not be included in the profile now, but be submitted to the Graphical Registration process along with other functional extensions. The reasoning is the same as for the larger number of functional extensions and registration proposals referred to above. For reference, the proposals are presented in the last section of this paper.

5.4 Encoding and File Format

The CALS AP raises the question of whether conforming interchange should be limited, as in the TOP AP, to the binary encoding. The CGM claims that the attributes of the three encodings are sufficiently unique that all three are justified. Implementation of all three encodings is considerably more of an effort than just the binary (which is the simplest in any case). Such a requirement could be a deterrent to acceptance of the CGM and the AP. There has not yet been a quantitative measurement of the differences between the encodings. The CALS AP recommends that such be done to justify or reject the specification of additional encodings in the next revision of the profile.

With regard to the physical format of the Binary metafiles - for the CALS profile, the weakly stated requirements of TOP AP section 6.2.9.2.2 are strengthened: the logical records shall be 80 octets in record-oriented environments; on magnetic tape, the records shall be blocked 10 at a time, forming 800 octet physical tape records.

In addition, the bit/octet/word order of section 4.3 of Part 3 of CGM (ANSI x3.122-1986) is the correct order for interchange between conforming CALS implementations.

5.5 Additional Requirements for Generators

In Section 6.2.6.6 of the TOP AP, restricted ranges are given for the values of primitive attributes. Implementation experience has showed that there is an ambiguity in connecting a "TOP Basic" CGM generator to an Application Programmer Interface (API) graphics system such as GKS. If the API layer passes attributes out of the Basic range, what should the generator do?

This CALS AP specifies that the values of attributes in 6.2.6.6, when passed to the generator outside of the Basic range, be handled as follows:

- If the index is selecting an attribute (e.g., linetype), then the CGM generator should map it by MODULO onto the Basic range;
- If the index is defining an attribute (e.g., color table), then it should be ignored if outside the Basic range.

These choices for the generator are consistent with Annex D of CGM, and with the reference to Annex D in clause 6.2.9 of the TOP AP.

In addition, it should be noted that the minimum data structure support of TOP AP section 6.2.9.1 are maximal limits for generators. CALS Basic CGM generators should not generate elements longer than those listed, because conforming CALS Basic interpreters are not required to accept longer primitives.

5.6 Additional Requirements for Interpreters

The TOP AP points to the specifications of Annex D of CGM for interpreters that do not have sufficient capability to accurately render the contents of a metafile. One of the goals of the CALS AP is to make the use of CGM completely unambiguous. This is not always required; for example, for preview of pictures black-and-white will often suffice where color has been specified. But at the time that a picture is to be incorporated into a document, the results must be completely predictable. An interpreter of such fidelity is going to be relatively expensive to produce and run, and is not needed all of the time.

Therefore the CALS AP determines that two conformance levels are needed for certifying interpreters:

1. preview-quality conformance - the fallback actions of Annex D of CGM may be used in some cases.
2. publication-quality conformance - the rendered picture must exactly correspond with the specification contained in the CGM - fonts must be correct, colors correct, ...

5.7 Specification of Semantic Ambiguities

The CGM leaves the semantics of a number of graphical details unspecified or "implementation dependent". This compromises predictable interchange. Although there will continue to be applications where it doesn't really matter, there are also those where it matters very much (technical illustrations for high-quality publications). These graphical details need to be addressed soon by additional functions, ESCAPES and GDPs. Some of these are already being specified and submitted for registration, as mentioned previously.

The CALS AP considered specifying the default behavior of these additional controls and attributes - items such as line join, cap, etc. Once again, doing so risks the possibility of being in conflict with standardized or registered elements. Specification of these details was thus deferred.

Clipping. Clipping shall be done to the intersection of the viewport and the device view surface limits when the clipping indicator is 'off'. Clipping shall be done to the intersection of the clip rectangle, the viewport and the device view surface limits when the clipping indicator is 'on'.

Linetype Continuation. Linetype shall be maintained (continued) across the interior vertices of a polyline.

Edge Type Continuation. Edge type shall be maintained (continued) across the vertices of a filled area boundary.

Line Cap. Deferred pending results of Graphical Registration.

Line Join. Deferred pending results of Graphical Registration.

6 Significant Issues Remaining for the CALS AP

The current specification of the CGM Application Profile for CALS has met most of its primary objectives. In particular, the profile is immediately usable, is a superset of the TOP profile, and reasonably implementable. The profile is viewed as an initial specification and is expected to grow and evolve to meet more advanced user requirements. There are several major issues that should be addressed before the next revision. They should be revisited and resolved in a year or less.

6.1 Exact Set of Functional Extensions

The paper by Carson presents a substantial list of extensions to CGM needed by CALS. This paper presents a couple more. The list of proposed extensions now includes:

1. a symbol library facility (under development in CGEM);
2. user defined linetype (2 different proposals);
3. user defined hatch style;
4. approximately 10 registered linetypes;
5. approximately 20 registered hatch styles;

6. several types of spline curves;

7. conics and conic arcs;

8. closed figure primitive;

9. arbitrary clipping boundary;

10. a number of registered fonts;

11. a completely new text model, via ESCAPEs and GDPs;

12. raster "input" and associated attributes for image processing;

13. additional line attributes, e.g., line cap, line join ...

More proposals may be forthcoming. Some of the collected proposals are likely to be controversial. It will be a significant task to sort through the proposals and registered items and adjust the profile.

6.2 Fonts

There are some more serious problems with the current state of the TOP and CALS APs - particularly the question of the Hershey fonts - than noted above. First, the Hershey fonts are not really fonts but are font/character set combinations. The motivation for their inclusion is that they are public domain, and some specification of fonts was desirable. The "CGM Extensions" study has pointed to a number of more useful and higher quality fonts. These should be considered for inclusion in an early revision of this profile.

Meanwhile, the Hershey fonts really require of the interpreter a certain number of typefaces and a certain number of character sets. The CALS profile states that the requirements of the CALS profile are served, and compatibility with the TOP profile is maintained, by requiring that the necessary character sets be supported in part, and the necessary typefaces be supported in part, so that the combinations required to render the listed 16 Hershey "fonts" are supported in full. Furthermore, "recognizably similar and of at least as high quality" is the criterion applied to font conformance. The exact (stroked) Hershey fonts then just become one way to realize this support.

The most serious problem in the area of fonts is that the Hershey fonts do not meet the needs of high quality illustration and publishing. Specification of a rich, useful and mandatory set of fonts for CALS is be a high priority future work item. The fonts of IGES are considered as starting point. However, the list in the "Needed CGM Extensions" report (of Carson) contains some proprietary names. It is undesirable to obligate conformers to the profile to have to pay commercial license fees. It may be possible to specify a useful set of fonts in terms of reference to font metrics, rather than reference to copyrighted names. This is a topic which CALS should pursue in the near future, before the next revision of the profile.

6.3 Encodings

There are suggestions that 2 or all 3 encodings should be conforming interchange - all 3 have distinct and valuable use. On the other hand, this would definitely increase the implementation burden for those adhering to the CALS profile. This question should be studied quantitatively, and experience with the binary profile accumulated, over the next year. It should then be determined if other encodings should be included in the next version of the profile.

6.4 3D CGM

The needs of the CALS constituency for 3D have not been much studied yet. IGES provides some 3D capability for product data, but is not a graphical metafile in the sense of CGM. Work is underway for CGM Addendum 2 (3D extensions) but will be a long time until completion. It is unclear whether CALS has immediate need for a 3D metafile, and whether whatever needs it might have are being addressed by the ISO 3D extension work. A clear reference model for CALS would help resolve the issue of the need for 3D.

7 Registration Proposals Developed as Part of the CALS AP Specification

The following two items were removed from the TOP AP draft. They were subsequently reformulated and are considered valuable functionality for the CALS community. It is recommended in the CALS AP that they be submitted for registration.

7.1 User Defined Linetype

This element defines a linetype and associates it with an index for future reference:

Parameters:

linetype (IX) - index of linetype being defined;
number of dash elements (I) - number of elements in the defined line pattern;
list of dash elements (nI) - I>=0, n>=1
n=1 means a solid line;
I=0 interpreted as a dot;
First element is a dash, second a space, etc;

Additional parameters (or ESCAPE attributes):

duty cycle unit selector = {VDC, mm, native device units,abstract} the value of 'abstract' indicates that the implementation may normalize and map the sum of the dash pattern elements at its discretion.

duty cycle (repeat length in units of '..selector')
These two controls define the length of the dash pattern.

adaptive flag = {no, yes} - an "adaptive" linetype is one where every vertex falls on an inked portion of the line. This is accomplished in plotters by temporarily modifying the duty cycle for each line segment (ceiling function) such that there is always an integral number of repeats (and all predefined linetypes have their gaps_array defined such that they begin and end with inked or "pen down" portions). Default is "no" or non-adaptive, so that the duty cycle is always the same regardless of line segment length, unless the user requests otherwise.

7.2 User Defined Hatch Style

This element defines a linetype and associates it with an index for future reference:

Parameters:

hatch index (IX) - index of hatch style being defined;
style indicator (E) - {parallel, crosshatch};
number of hatch elements (I) - number of elements in
    the defined hatch style;
list of hatch elements (nI) - I>=0, n>=2
    the array gives alternating (line width, gap width) - a direct analogy to the linetype array. Center of the first hatch line is matched up with PATTERN REFERENCE POINT, if implemented. 0 interpreted as thinnest line width available. Error if sum of hatch elements is 0.

Additional parameters (or ESCAPE attributes):

units indicator = {VDC, mm, device units, abstract}

specifies the units in which 'angle' and 'duty cycle length' are specified. Also controls the manner of transformation of the hatching: If VDC, then the hatching transforms with segment transform and anisotropic transforms (as if you had done POLYLINES); otherwise, the hatching is like "wallpaper" that shows through the polygon-shaped hole - you've mapped all that's necessary into device units and are doing hatching in device space. The value of 'abstract' indicates that the implementation may normalize and map the sum of the dash pattern elements at its discretion.

angle (dx, dy)* - default is horizontal;

duty cycle (repeat length)
    Specifies the distance measured perpendicular to the hatch line. The sum of hatch elements in the hatch element list is normalized to this distance before presentation of the hatch on the view surface.