PROCEDURE PortFrameSender : Combinational Function
PULSEDINPUTVAR FROM SegmentSendO : pseg;
STATICINPUTSET FROM ResetSender : ok, ng;
BEGIN
VAR pf: PortFrame;
IF ResetSender.State = PendSend THEN
BEGIN
pf := ((DestinationPortAddress, OriginPortAddress), 'Reset');
SEND (DestinationRouterAddress, pf) TO PortFrameSendO;
SEND 'Sent' TO ResetSender;
END;
ELSE
BEGIN
WHILE (ResetSender.Output = ok) AND FlowControl.Output = ok) DO
BEGIN
Npl := LastNpSent.Output + 1;
Npph := NpSend.Output;
pf.pseg := (DestinationPortAddress, OriginPortAddress,
Np, Nph, ...);
SEND (DestinationRouterAddress, pf) TO
PortFrameSendO;
SEND 'Sent' TO LastNpSent;
END;
END;
END;

Figure 4. A programming language description of a port frame
sender.

The specification of computer graphics systems

Based on the specification techniques discussed in the
two previous sections and on the earlier discussion of the
desirable properties of a formal specification, I propose
that a formal specification for computer graphics systems
comprise descriptions of

- the interface between graphics and the host language,
and graphics and a graphical display device;
- the structure of the graphics system; and
- the functions performed by the graphics system.

Interface. I recommend that both of the required inter-
face specifications be written using a message-based
model and abstract data types. As shown earlier, such
an interface is easily bound to a standard procedure
(subroutine) call mechanism and can also support more
distributed models of computation. Sproull's technique
for specifying an abstract procedure call can be general-
ized to define such a message-based interface.

Structure. The structure of the graphics environment
should be specified using both simple state models (simi-
lar to Figure 2) and block diagrams (similar to Figure 3)
as appropriate. These should be augmented by a formal-
ly defined system state vector. The items in the state vec-
tor should be instances of abstract data types defined like
those discussed by Guttag and Horning.9 The block
diagrams should be refined to such a level of detail that
the final blocks correspond to the graphical functions in
the next lower level of specification.

Functions. The specification of graphical functions
should correspond to the "routine" specifications of
Guttag and Horning. The functions to be specified will
probably be, for the most part, the same as the functions
in present informal standards (Polyline-Abs-2, Set-
Color-Index, etc.). Although as a first approximation
such functions might be specified in a standard program-
ing language such as Pascal, as suggested by Rosen-	hal,6 the final formal specification should be written
more abstractly.

After much experimentation with various techniques
for specifying the semantics of graphical functions, the
ANSI committee settled on the use of ordinary mathemat-
ics for writing specifications. We selected this technique
over the Dijkstra-style predicate transformers used by
Guttag and Horning for several reasons:

specification produced by Piatkowski10 for the OSI
reference model.

Additional details on the design and specification of
communication protocols are available from a number
of sources.8,10,22-24