

2

AD-A261 732



DTIC
ELECTE
MAR 24 1993
S C D

THREE VIEW LANGUAGE: A GRAPHICS-BASED LANGUAGE FOR REQUIREMENTS DEFINITION

SPC-92114-CMC

VERSION 01.00.00

MARCH 1993

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

98 3 24 55

93-06115



THREE VIEW LANGUAGE: A GRAPHICS-BASED LANGUAGE FOR REQUIREMENTS DEFINITION

SPC-92114-CMC

VERSION 01.00.00

MARCH 1993

Dr. Paul Ward

This material is based in part upon work sponsored by the Defense Advanced Research Projects Agency under Grant # MDA972-92-J-1018. The content does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred.

This document accompanies a videotape of the same presentation recorded live at the Software Productivity Consortium in February 1992. It is recommended that the videotape be viewed with these viewgraphs at hand.

This material, including slides copyrighted by Dr. Ward, may be reproduced without restriction. The right to reproduce this material without restriction has been previously granted by Dr. Ward.

Produced by the
SOFTWARE PRODUCTIVITY CONSORTIUM

under contract to the
VIRGINIA CENTER OF EXCELLENCE
FOR SOFTWARE REUSE AND TECHNOLOGY TRANSFER
SPC Building
2214 Rock Hill Road
Herndon, Virginia 22070

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

**SOFTWARE
DEVELOPMENT
CONCEPTS**

Date: December 17, 1992

TO: SOFTWARE PRODUCTIVITY CONSORTIUM
2214 Rock Hill Road
Herndon, VA 22070

ATTN: Technology Transfer Clearinghouse

This letter is in reference to a talk given on July 27, 1992, by Paul T. Ward at the Software Productivity Consortium ("SPC"), on the subject of "3VL: A Formal Description of a Graphics-based, Three-view Operational Specification Language".

With our permission, SPC has produced a videotape of this material. The accompanying paper version is owned and copyrighted by Software Development Concepts. As an authorized recipient of this videotape and papercopy, sent to you by SPC, we hereby grant you permission to reproduce and distribute our copyrighted material, provided that when such reproduction and distribution occurs by the Software Productivity Consortium you include a statement that it is for internal purposes within the recipient's organization.

The paper copies of the presentation material must be marked with our Copyright Notice.

Questions regarding 3VL can be directed to the developer, Paul T. Ward, care of Software Development Concepts (contact info below).

Yours Truly,



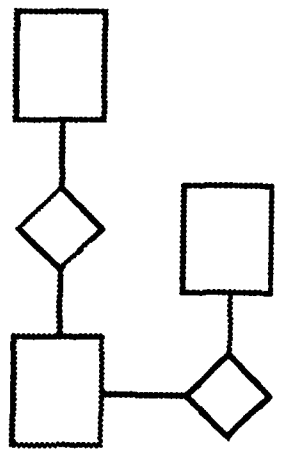
Pamela M. Plate
Managing Partner

3VL: A Graphics-Based Language for Requirements Definition

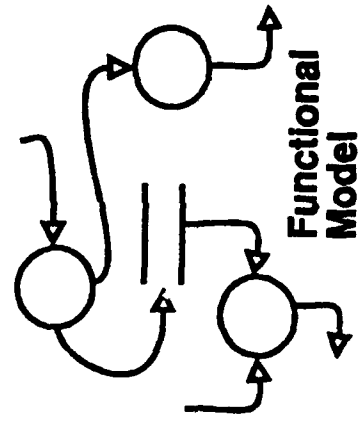
Paul T. Ward

**Copyright © 1992 by
Software Development Concepts
185 West End Avenue, Suite 27-L
New York, NY 10023
Phone 212-362-1391**

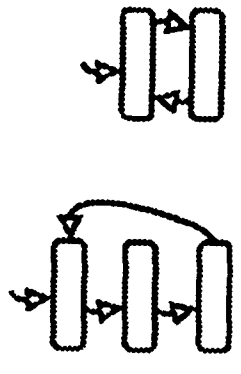
Many developers use combinations of notations to build models representing various views of the requirements for a software system.



Data Model



Functional Model



Behavioral Model

These models are often verified by execution.



Since the mid-1980s:

- **Formal execution rules (operational semantics) have been defined for some of these notations.**
- **A few CASE tools (notably Statemate) have provided for automated execution of graphics-based requirements definition models.**

However, there are still barriers to building comprehensive, executable, graphics-based requirements definition models:

- **Existing execution rules for functional (data flow) notations are superficial.**
- **There are no execution rules for data modeling (entity-relationship) notations.**
- **There is no provision for dynamic instantiation.**

3VL is an extension of the Harel statechart-activity chart notation which provides:

- **A comprehensive operational semantics encompassing the data, functional, and behavioral views**
- **dynamic creation and deletion of objects**

3VL is also designed explicitly for use on an interactive electronic display medium; extensive use is made of colors and textures to aid comprehension.

3VL reduces the number of diagrams required to build a model by providing:

- unlimited nesting within a single diagram**
- a layered notation from which views are obtained by selective suppression of details**

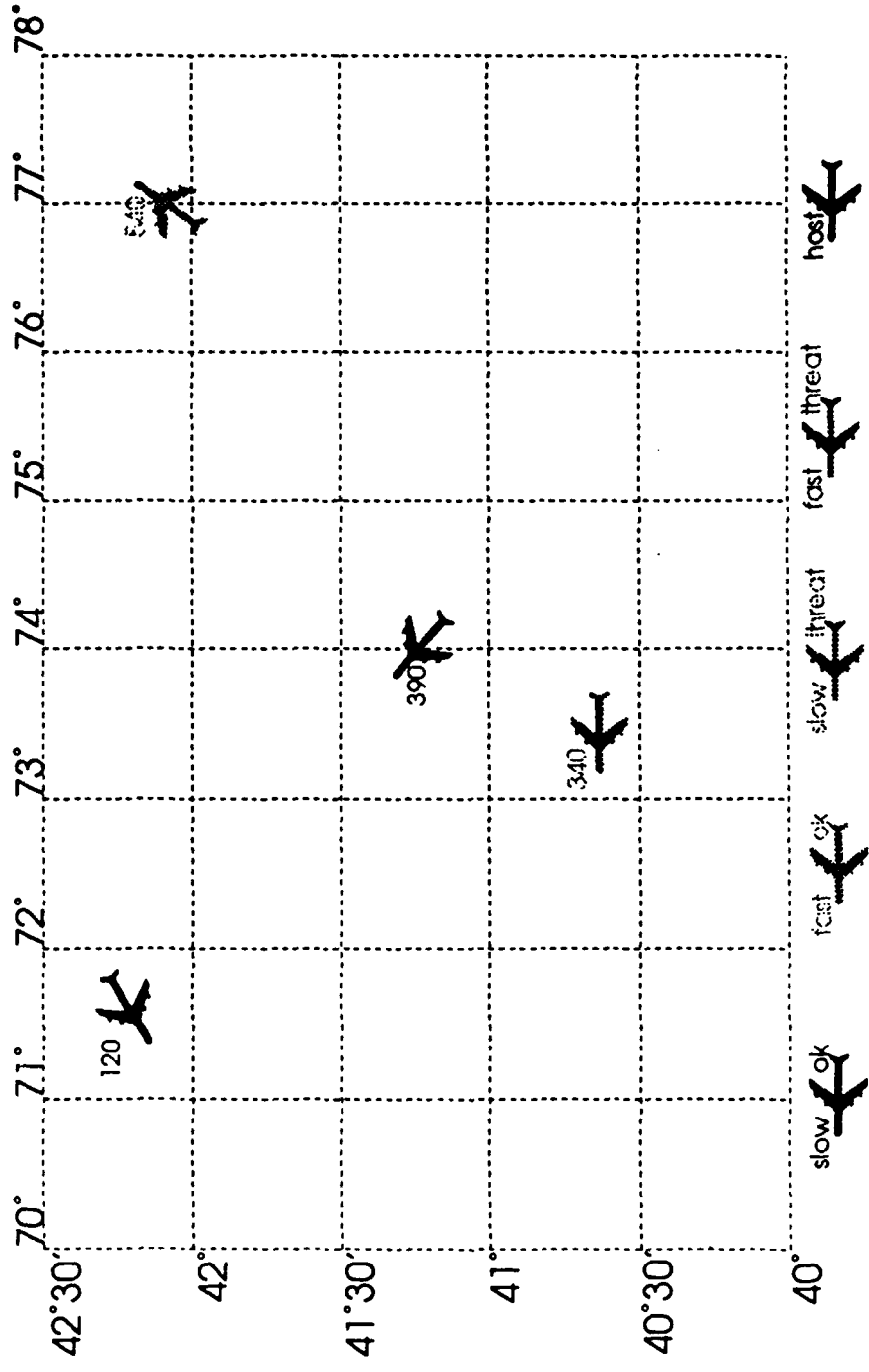


SAMPLE PROBLEM

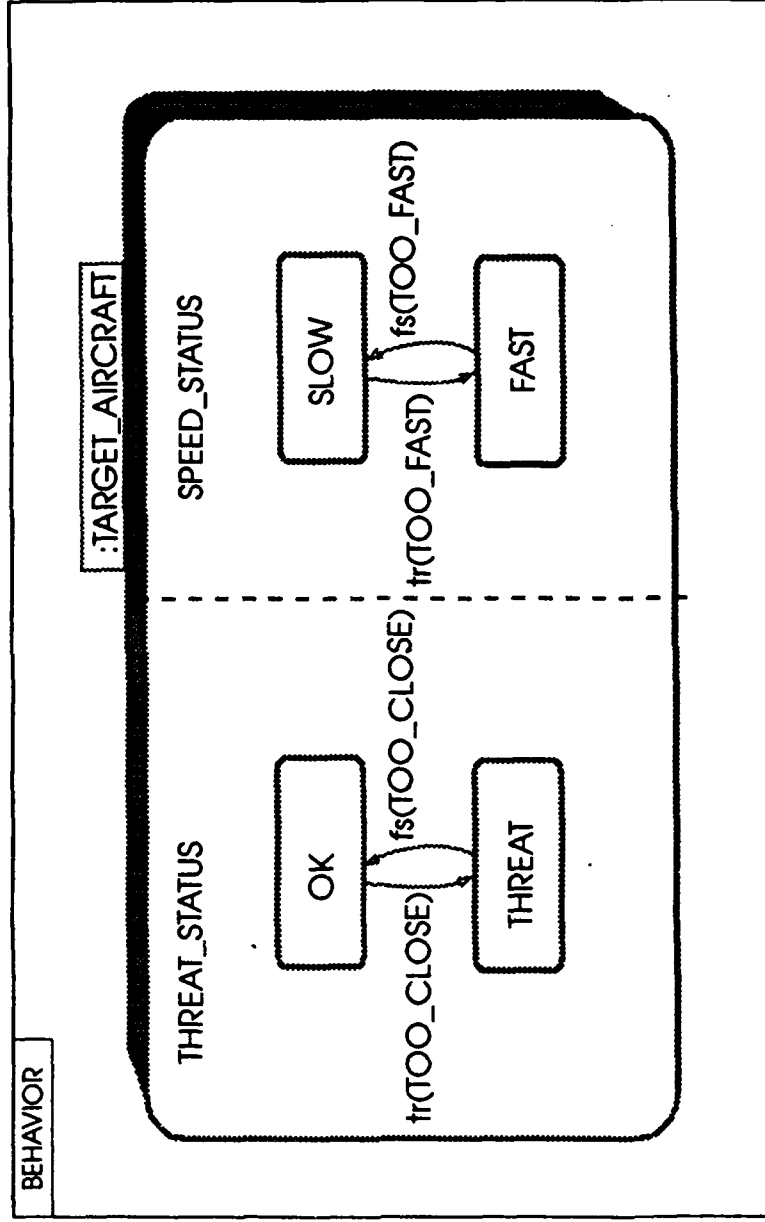
A Collision Warning system tracks the coordinates of a host aircraft and of target aircraft in its vicinity, displaying these coordinates and also the threat (if any) posed by the target aircraft.

Coordinates of a target aircraft that poses a threat are sent to a collision avoidance system.

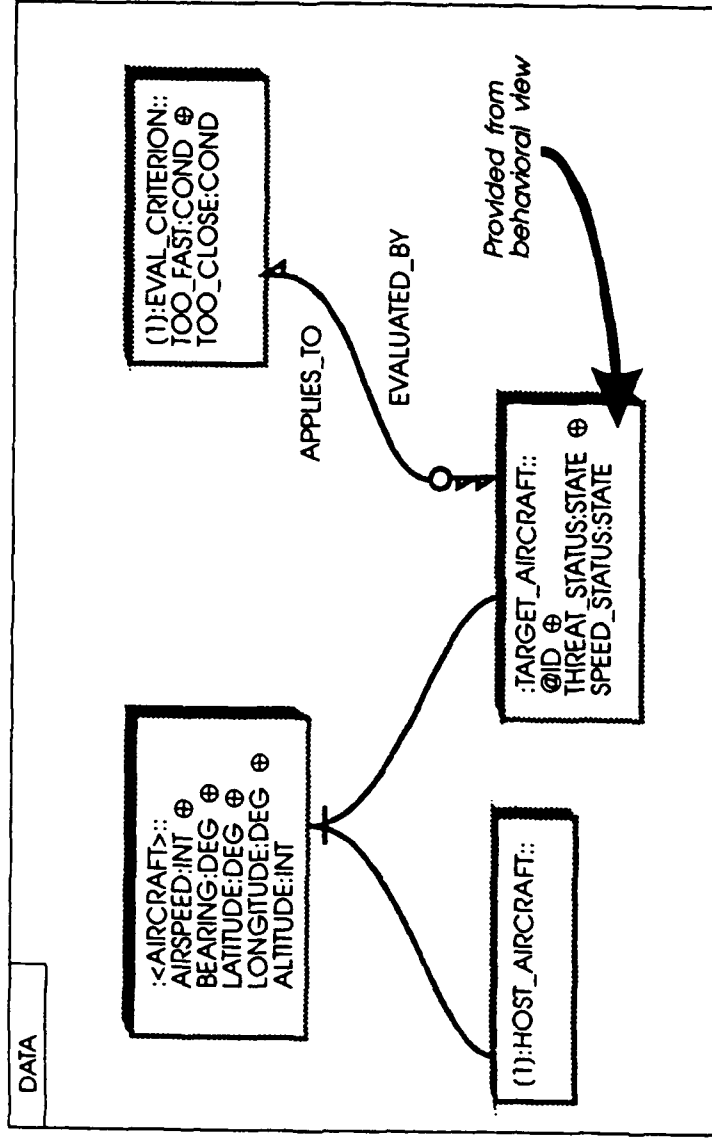
Parameters for threat determination may be adjusted dynamically.



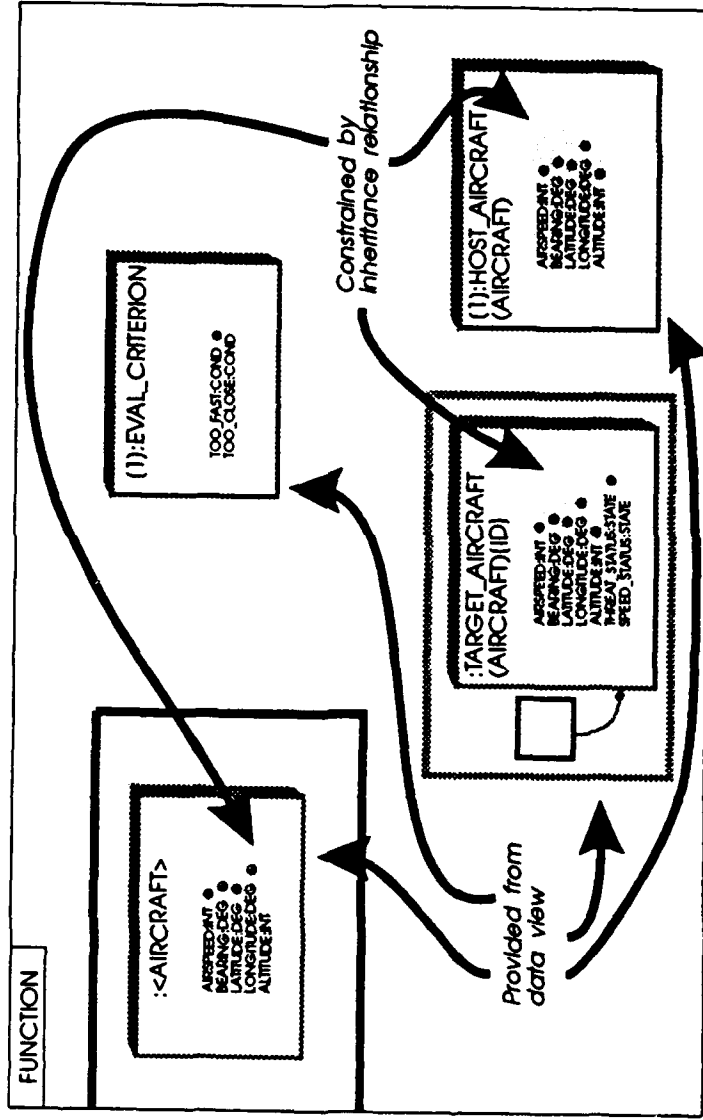
Statechart notation is used for the behavior view.



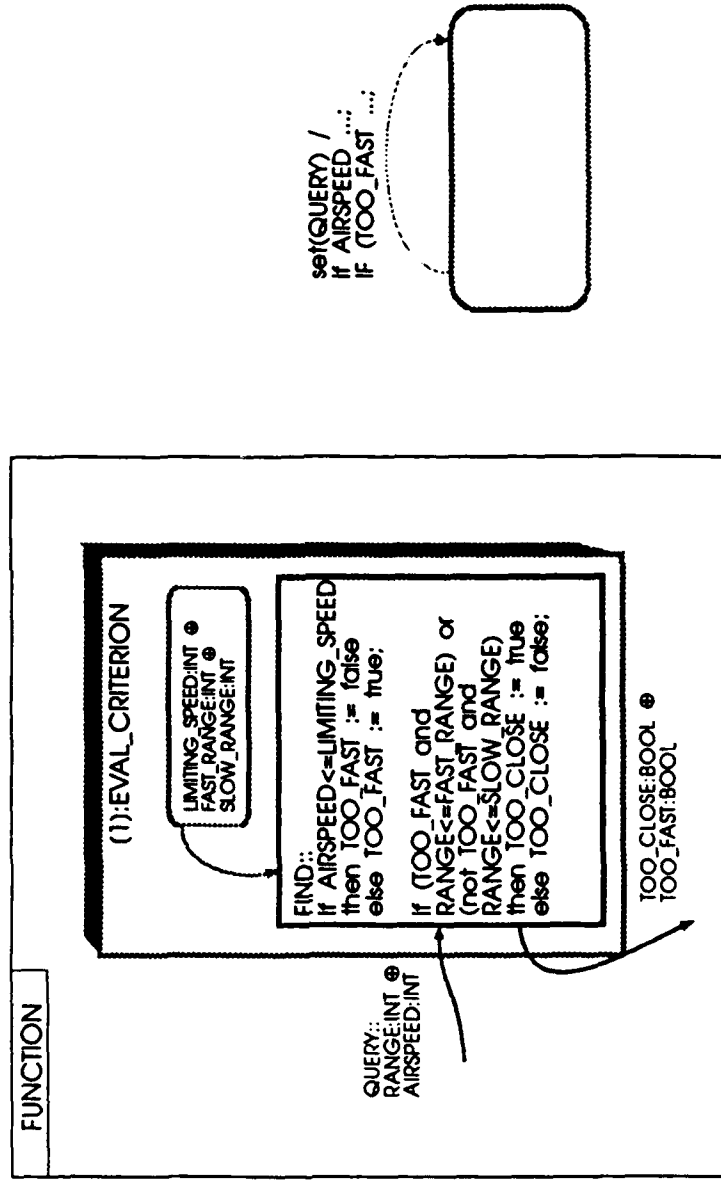
Entity-relationship notation is used for the data view.



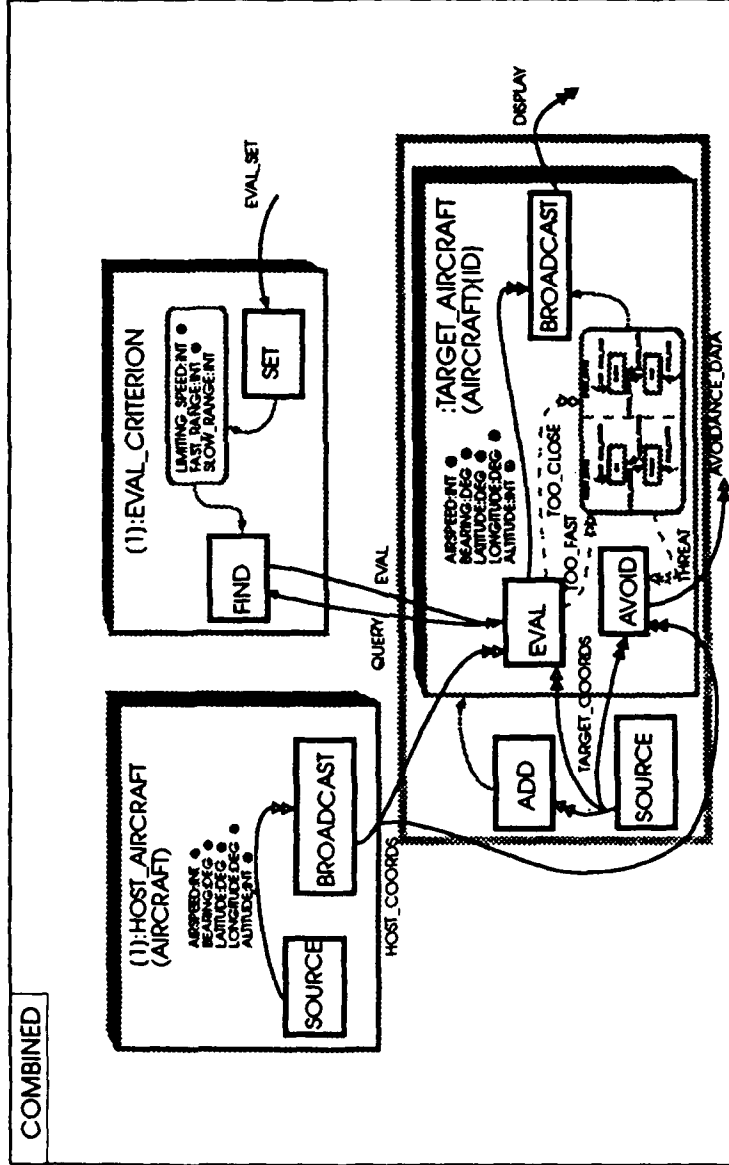
Activity charts are used for the functional view.



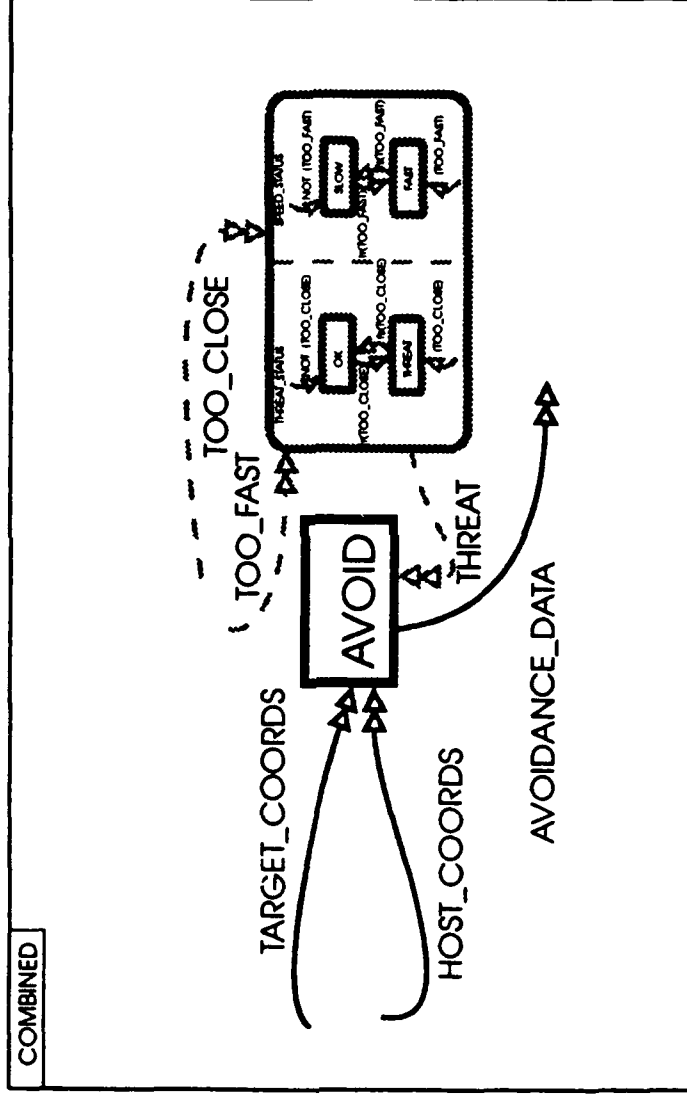
The definition of an activity is equivalent to a statechart transition expression.



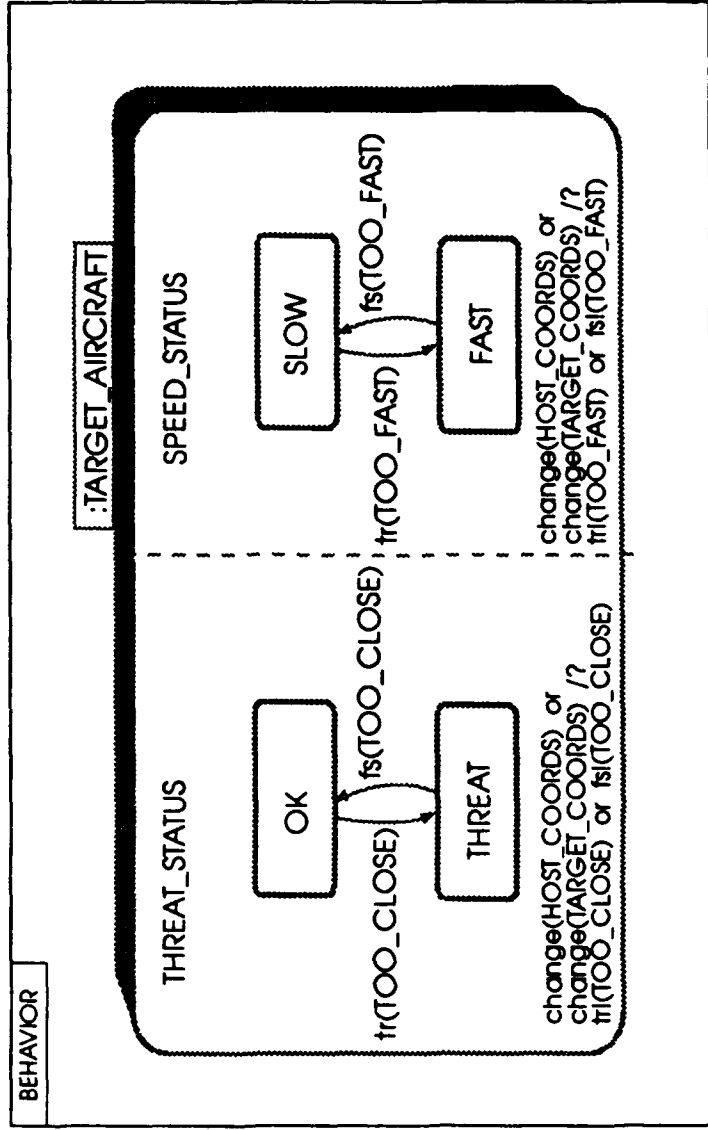
The 3 notations are layered into a combined view.



Control of activities by statecharts is shown by labeled flow connections.



In data and behavior views, details of functional connections are replaced by assertions.



WORK IN PROGRESS

- recording, browsing, and reuse of class definitions**
- Construction of units representing sets of 3VL programs, and extraction of legal programs from these units**
- translating 3VL requirements models into design models**

Questions or comments on content should be directed to:

**Dr. Paul T. Ward
Software Development Concepts
185 West End Avenue, Suite 27-L
New York, NY 10023
(212) 362-1391**

Or to:

**Dr. Stuart Faulk
Software Productivity Consortium
2214 Rock Hill Road
Herndon, VA 22070
(703) 742-7117**

***Send feedback on the Consortium's Video Program and
orders for video products to:***

**Technology Transfer Clearinghouse
Software Productivity Consortium
2214 Rock Hill Road
Herndon, VA 22070
(703) 742-7211**