Application of Symbolic Processing to Command and Control: An Advanced Information Presentation System

Annual Technical Report No. 2
Volume III - Program Source Files

F. Zdybel, A. Gibbons, N. Greenfeld, and M. Yonke
Frank, Jeff, Norton, Martin

10 July 1978

Prepared for:
Defense Advanced Research Projects Agency

This document has been approved for public release and sale; its distribution is unlimited.
This report describes the work performed in the second year of the three-year contract to explore the application of symbolic processing to command and control (C2); specifically, the graphics interface between the C2 user and a complex C2 decision support system. In Volume I, the goals and approaches used in the design of the prototype system, AIPS (Advanced Information Presentation System), are discussed, as
well as the year's efforts to extend the prototype. An overview of the current AIPS system is also provided.

Volume II contains the complete AIPS knowledge base. This document provides the fully-inherited structure that the system sees during operation.

Volume III contains the programs that manipulate the knowledge base and provide the active behavioral component of the system.
APPLICATION OF SYMBOLIC PROCESSING TO COMMAND AND CONTROL:
AN ADVANCED INFORMATION PRESENTATION SYSTEM

Frank Zdybel, Jeff Gibbons, Norton Greenfeld and Martin Yonke

August 1981
Annual Technical Report #2
Volume 3 - Program Source Files

Prepared by:
Bolt Beranek and Newman Inc.
10 Moulton Street
Cambridge, Massachusetts 02238

Prepared for:
Defense Advanced Research Projects Agency
Information Processing Techniques Office
1400 Wilson Boulevard
Arlington, Virginia 22209
This research has been supported by the Defense Advanced Research Projects Agency under Contract N00039-79-C-0316, ARPA Order No. 3740, and monitored by the Naval Electronics System Command (Program Code 9D30).

The views and conclusions contained in this paper are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.
1. FUNCTION INDEX

The following contains an index to all functions in the AIPS source files. These files are generally organized as follows:

- Miscellaneous
  - AIPS: a file that merely loads all other files in the system
  - AIPSTEMP: a file of useful but otherwise uncategorizable functions
  - AIPSUTILITY: a file of utility routines

- Presentation
  - AIPSDISPLAY: the main presentation routines
  - AIPSUCS: definition of coordinate systems
  - AIPSMAPI: definition of the "map" presentation format
  - AIPSTABLE: definition of the "table" presentation format

- Realization
  - AIPSREALIZATION: the main realization modules
  - AIPSTODRAW: interface to the graphics primitives
  - AIPSDEVICES: a model of the graphics devices available
  - AIPSWINDOW: a simple window package in AIPS knowledge structures

- Domain Model
  - AIPSODOWN: the upper-level shared domain world structure
  - AIPSTPS: general time system knowledge
  - AIPSUDEOOGRAPI: general shared geographic knowledge
  - AIPSNNAVAL: knowledge for depicting naval data
  - AIPSNTDS: specific knowledge for the NTDS presentation format
-- Index to Files --

<table>
<thead>
<tr>
<th>File Name</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NEWAIPS&gt;AIPS..3</td>
<td>8-Nov-80 10:49pm</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPSCS..10</td>
<td>3-Jul-80 5:03pm</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPSDEVices..20</td>
<td>19-Jan-81 1:23am</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPSDISPLAY..47</td>
<td>19-Jan-81 10:43pm</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPSDOMAIN..5</td>
<td>10-Dec-80 11:41pm</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPS GEOGRAPHY..9</td>
<td>11-Dec-80 12:01am</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPS MAP..57</td>
<td>13-Nov-80 3:55am</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPSNAVAL..3</td>
<td>17-Jul-80 2:14am</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPS NTDS..26</td>
<td>13-Nov-80 2:21am</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPS REALIZATION..28</td>
<td>29-Jan-81 1:10am</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPSTEMAP..36</td>
<td>18-Dec-80 8:03am</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPSTEMP..111</td>
<td>2-Feb-81 3:43am</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPSTODRAW..63</td>
<td>2-Feb-81 3:12am</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPSTS..2</td>
<td>10-Jul-80 11:58pm</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPS UTILITY..9</td>
<td>19-Dec-80 12:34am</td>
</tr>
<tr>
<td>&lt;NEWAIPS&gt;AIPS WINDOW..122</td>
<td>2-Feb-81 1:15am</td>
</tr>
</tbody>
</table>

AIPSGetSpecifiedMetaDescriptions..........................AIPSDISPLAY 12
CollectTemplatesDescribingEntity...........................AIPSDISPLAY 13
DisplaySurfac ForGroundP.................................AIPSTODRAW 27
EstablishDomainEntityPositionInCS.........................AIPS MAP 23

FindApplicableApplicationSlots.......................AIPSDISPLAY 14
FindApplicationObjects..............................AIPSDISPLAY 15
FindColumnTableRows..................................AIPSTABLE 10
FindColumnWidth......................................AIPSTABLE 11
FindCostOfDisplay....................................AIPSDISPLAY 16
FindCoveringlntendedApplication........................AIPSDISPLAY 17
FindDistanceSquaredFromDisplayForm.................AIPSDISP 46
FindDomainCoordinateSystem..........................AIPSMAP 8
FindDrawingContextsForDisplay........................AIPSTODRAW 28
FindMapEntitiesFromApplication........................AIPSMAP 9
FindMapItemPrototypes................................AIPSMAP 21
FindMappingBetweenCSs.................................AIPSMAP 22
FindMostSpecializedConcept..........................AIPSTEMP 17
FindRoleValuesTransitively..........................AIPSTODRAW 29
FindRowHeight.........................................AIPSTABLE 12
FindRowTableColumns.................................AIPSTABLE 13
FindScaleUnitDistance.................................AIPSMAP 10
FindTableEntityIndicators............................AIPSTABLE 14
FindTableThemeElementIndicators.....................AIPSTABLE 15
FindTableThemeElementlndicatorsAndOrientation......AIPSTABLE 16
FindTemplateObjects..................................AIPSDISPLAY 18
FindTemplateSlots...................................AIPSDISPLAY 19
FindTopLevelDisplaysOnViewSurface....................AIPSTODRAW 30
<table>
<thead>
<tr>
<th>Function Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetFontCharacterDef</td>
<td>31</td>
</tr>
<tr>
<td>GuageTemplateRGCoverageOfTemplate</td>
<td>20</td>
</tr>
<tr>
<td>HomogenousConceptListP</td>
<td>18</td>
</tr>
<tr>
<td>Init2DCOORDINATESYSTEM</td>
<td>1</td>
</tr>
<tr>
<td>Init2DLINEARTRANSFIX</td>
<td>2</td>
</tr>
<tr>
<td>Init2DLINEARTRANSFORM</td>
<td>3</td>
</tr>
<tr>
<td>InitAIPS</td>
<td>1</td>
</tr>
<tr>
<td>InitAPERTURE</td>
<td>1</td>
</tr>
<tr>
<td>InitARRANGEMENT</td>
<td>1</td>
</tr>
<tr>
<td>InitCARTESEANSYSTEM</td>
<td>4</td>
</tr>
<tr>
<td>InitCIRCLE</td>
<td>1</td>
</tr>
<tr>
<td>InitCLOSEDCURVE</td>
<td>2</td>
</tr>
<tr>
<td>InitCOLUMN</td>
<td>2</td>
</tr>
<tr>
<td>InitCOLUMNTABLE</td>
<td>3</td>
</tr>
<tr>
<td>InitCommonDomainModel</td>
<td>1</td>
</tr>
<tr>
<td>InitCONRACSCREEN</td>
<td>1</td>
</tr>
<tr>
<td>InitCOORDINATESYSTEM</td>
<td>6</td>
</tr>
<tr>
<td>InitCoordinateSystemConcepts</td>
<td>6</td>
</tr>
<tr>
<td>InitCTYWINDOW</td>
<td>2</td>
</tr>
<tr>
<td>InitCURVE</td>
<td>3</td>
</tr>
<tr>
<td>InitDeviceConcepts</td>
<td>3</td>
</tr>
<tr>
<td>InitDevices</td>
<td>4</td>
</tr>
<tr>
<td>InitDISPLAY</td>
<td>3</td>
</tr>
<tr>
<td>InitDISPLAYABSTRACTION</td>
<td>4</td>
</tr>
<tr>
<td>InitDISPLAYATOM</td>
<td>4</td>
</tr>
<tr>
<td>InitDISPLAYCOMPOSITE</td>
<td>5</td>
</tr>
<tr>
<td>InitDISPLAYFORM</td>
<td>6</td>
</tr>
<tr>
<td>InitDisplayFormConcepts</td>
<td>7</td>
</tr>
<tr>
<td>InitDISPLAYITEM</td>
<td>3</td>
</tr>
<tr>
<td>InitDISPLAYITEMABSTRACTION</td>
<td>2</td>
</tr>
<tr>
<td>InitDISPLAYSURFACE</td>
<td>4</td>
</tr>
<tr>
<td>InitDISTANCE</td>
<td>2</td>
</tr>
<tr>
<td>InitDISTANCEUNIT</td>
<td>3</td>
</tr>
<tr>
<td>InitEDGESET</td>
<td>8</td>
</tr>
<tr>
<td>InitELLIPSE</td>
<td>9</td>
</tr>
<tr>
<td>InitENEMYNTDSPLANE</td>
<td>1</td>
</tr>
<tr>
<td>InitENEMYNTDSSHIP</td>
<td>2</td>
</tr>
<tr>
<td>InitENEMYNTDSSUB</td>
<td>3</td>
</tr>
<tr>
<td>InitENEMYPLANE</td>
<td>1</td>
</tr>
<tr>
<td>InitENEMYSHIP</td>
<td>2</td>
</tr>
<tr>
<td>InitENEMYSUB</td>
<td>3</td>
</tr>
<tr>
<td>InitFASTWINDOW</td>
<td>2</td>
</tr>
<tr>
<td>InitFRIENDLYNTDSPLANE</td>
<td>4</td>
</tr>
<tr>
<td>InitFRIENDLYNTDSSHIP</td>
<td>5</td>
</tr>
<tr>
<td>InitFRIENDLYNTDSSUB</td>
<td>6</td>
</tr>
<tr>
<td>InitFRIENDLYPLANE</td>
<td>4</td>
</tr>
<tr>
<td>InitFRIENDLYSHIP</td>
<td>5</td>
</tr>
<tr>
<td>InitFRIENDLYSUB</td>
<td>6</td>
</tr>
<tr>
<td>InitGeographyModel</td>
<td>2</td>
</tr>
</tbody>
</table>
InitGEOREGION..................................AIPSGEOGRAPHY 1
InitHIDDENPLANE................................AIPSDSVICES 5
InitHOMOGENEOUSREGION..........................AIPSTEMP 5
InitINTEGERSYSTEM..............................AIPSCS 7
InitINTERVAL..................................AIPSDOMAIN 4
InitINTERVALUNIT...............................AIPSDOMAIN 5
InitINVISIBLESURFACE...........................AIPSTEMP 6
InitTEMPLATE..................................AIPSDISPLAY 5
InitLANDFEATURE................................AIPSGEOGRAPHY 3
InitLANDMASS..................................AIPSGEOGRAPHY 4
InitLINEARTRANSFORM............................AIPSCS 8
InitLINESEGMENT.................................AIPSMAPREALIZATION 10
InitMAP..........................................AIPSMAP 1
InitMapConcepts................................AIPSMAP 6
InitMAPGRID....................................AIPSMAP 2
InitMAPITEM....................................AIPSMAP 3
InitMAPLEGEND..................................AIPSMAP 4
InitMAPPING.....................................AIPSCS 9
InitMAPWITHBACKGROUND.........................AIPSMAP 5
InitMETRIC......................................AIPSDOMAIN 6
InitMiddleEarth................................AIPSTEMP 8
InitMOTILEDISPLAYITEM........................AIPSTEMP 7
InitNATION.....................................AIPSGEOGRAPHY 5
InitNavalDomainModel..........................AIPSNAVAIAL 7
InitNONSCROLLWINDOW..........................AIPSWINDOW 3
InitNTDSConcepts...............................AIPSNSTS 7
InitNTDSENGAGEMENTMARK........................AIPSNSTS 8
InitNTDSTEMIT.................................AIPSNSTS 9
InitNTDSMART.................................AIPSNSTS 10
InitNTDSVELOCITYLEADER........................AIPSNSTS 11
InitORTHOGONALSYSSTEM.........................AIPSCS 10
InitPHYSOBJECT...............................AIPSDOMAIN 7
InitPLACE......................................AIPSDOMAIN 8
InitPLANE......................................AIPSNAVAIAL 8
InitPLANESURFACE..............................AIPSTEMP 9
InitPLATFORM..................................AIPSNAVAIAL 9
InitPOINT......................................AIPSMAPREALIZATION 11
InitPOLYGON....................................AIPSMAPREALIZATION 12
InitPOSITION..................................AIPSDOMAIN 9
InitPRESENTATION..............................AIPSDISPLAY 2
InitPresentationModel........................AIPSDISPLAY 1
InitRealizationModel..........................AIPSMAPREALIZATION 15
InitRECTANGLE..................................AIPSMAPREALIZATION 13
InitRECTANGULARMAP............................AIPSMAP 7
InitRECTANGULARREGION........................AIPSTEMP 10
InitREGION.....................................AIPSDOMAIN 10
InitREGULARPOLYGON............................AIPSMAPREALIZATION 14
InitROW........................................AIPSTABLE 4
InitROWTABLE..................................AIPSTABLE 5
InitSCROLLWINDOW..............................AIPSWINDOW 4
InitSHIP.........................................AIPSNAVAIAL 10
InitSLOWWINDOW.................................. AIPSWINDOW 5
InitSPEED.......................................... AIPSDOMAIN 11
InitSPEEDUNIT.................................... AIPSDOMAIN 12
InitSTABLEDISPLAYITEM......................... AIPSTEMP 11
InitSUB........................................... AIPSNAVAL 11
InitTABLE.......................................... AIPSTABLE 6
InitTableConcepts................................ AIPSTABLE 9
InitTEMPLATE....................................... AIPSDDISPLAY 6
InitTEXT........................................... AIPSDDISPLAY 7
InitTEXTCOLUMN................................... AIPSTABLE 7
InitTEXTROW....................................... AIPSTABLE 8
InitTIME........................................... AIPSDOMAIN 13
InitTIMESYSTEM................................... AIPSTS 1
InitTimeSystemConcepts.......................... AIPSTS 2
InitTRANSFIX....................................... AIPSCS 11
InitTRANSLATION................................... AIPSCS 12
InitTTYWINDOW...................................... AIPSWINDOW 6
InitTURTLESSCREEN................................ AIPSDOMAIONS 6
InitUNIT........................................... AIPSDOMAIN 14
InitUNKNOWNNTDSPAN................................ AIPSNTDS 12
InitUNKNOWNNTDSSHIP.............................. AIPSNTDS 13
InitUNKNOWNNTDSSUB.............................. AIPSNTDS 14
InitUNKNOWNPLANE................................. AIPSNAVAL 12
InitUNKNOWNSHIP................................... AIPSNAVAL 13
InitUNKNOWNSUB.................................... AIPSNAVAL 14
InitUSERMODEL..................................... AIPSTEMP 12
InitUserModel..................................... AIPSTEMP 13
InitUT............................................... AIPSTS 3
InitVEHICLE........................................ AIPSDOMAIN 15
InitViewingOrganizationModel.................... AIPSTEMP 16
InitVIEWSURFACE.................................. AIPSTEMP 14
InitVIEWSYSTEM.................................... AIPSCS 13
InitVISIBLESURFACE............................... AIPSTEMP 15
InitWATERBODY..................................... AIPSGEOGRAPHY 6
InitWATERFEATURE................................. AIPSGEOGRAPHY 7
InitWINDOW......................................... AIPSWINDOW 7
InitWindowConcepts............................... AIPSWINDOW 8
InitWORLDPOSITION............................... AIPSGEOGRAPHY 8

LDIFFERENCE.................................... AIPSTO DRAW 32

MakeColumnString................................ AIPSDISPLAY 21
MakeDefaultTableGround......................... AIPSTABLE 17
MakeDefaultTextString........................... AIPSDISPLAY 8
MakeDisplayName.................................. AIPSDISPLAY 9
MakeGroundForMapWithBackground................ AIPSMAP 11
MakeLegendApplication........................... AIPSMAP 12
MakeLinearMapTransformAndGroundFromEntities... AIPSMAP 13
MakeLinearMapTransformFromGroundAndEntities.. AIPSMAP 14
MakeMapItemApplication......................... AIPSMAP 24
MakeMapItemLabel?................................. AIPSMAP 15
Function Index

MakeMapItems.................................... AIPSMAP 16
MakeMapLabel.................................... AIPSMAP 17
MakeMapTransformForMapWithBackground........... AIPSMAP 18
MakeNTDSEngagementMark?.......................... AIPSNMTDS 15
MakeNTDSGroupStrengthMark?....................... AIPSNMTDS 16
MakeNTDSVelocityLeader?........................ AIPSNMTDS 17
MakeRectangle................................ AIPSMAP 19
MakeSimpleMapLegend............................. AIPSNMTDS 20
MakeTableColumnOrderList......................... AIPSTABLE 18
MakeTableEntryColumns........................... AIPSTABLE 19
MakeTableEntryRows.............................. AIPSTABLE 20
MakeTableFoot?................................ AIPSTABLE 21
MakeTableHeaderRow............................. AIPSTABLE 22
MakeTableLeftIndex............................. AIPSTABLE 23
MakeTableRightIndex?............................. AIPSTABLE 24
MakeTableRowOrderList.......................... AIPSTABLE 25
MakeTextHeight.................................. AIPSTABLE 26
MapPositionDescription........................ AIPSTABLE 27
OpenWindow........................................ AIPSWINDOW 9
RotatePointsAroundPoint........................ AIPSTODRAW 34
ROUND............................................ AIPSTODRAW 33
RoundPoint....................................... AIPSTODRAW 35
RoundPoints..................................... AIPSTODRAW 36
RptCoord......................................... AIPSGEOGRAPHY 10
ScrollFastWindow................................. AIPSTODRAW 12
ScrollSlowWindow................................ AIPSTODRAW 13
ScrollTTYWindow................................ AIPSTODRAW 3
ScrollWindow..................................... AIPSTODRAW 37
SelectDefaultDisplay............................ AIPSTODRAW 22
SelectDisplay.................................... AIPSTODRAW 23
SelectFromAmongSufficientDisplays.............. AIPSTODRAW 24
SendMessage..................................... AIPSTODRAW 25
SetRoleValues.................................... AIPSTODRAW 26
SupplyConceptMetaIndicator....................... AIPSTODRAW 25
SupplyRoleMetaIndicator........................ AIPSTODRAW 26
TDCircle......................................... AIPSTODRAW 4
TDCurve.......................................... AIPSTODRAW 5
TDDisplay........................................ AIPSTODRAW 6
TDDisplayComposite.............................. AIPSTODRAW 7
TDDisplayItemAbstraction......................... AIPSTODRAW 8
TDDisplaySurface................................. AIPSTODRAW 9
TDEdgeSet....................................... AIPSTODRAW 10
TDEllipse........................................ AIPSTODRAW 11
TDFastWindow.................................... AIPSTODRAW 12
TDIWFastWindow................................ AIPSTODRAW 13
<table>
<thead>
<tr>
<th>Function Index</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDIWNonScrollWindow</td>
<td>AIPSTODRAW 14</td>
</tr>
<tr>
<td>TDIWSlowWindow</td>
<td>AIPSTODRAW 15</td>
</tr>
<tr>
<td>TDLineSegment</td>
<td>AIPSTODRAW 16</td>
</tr>
<tr>
<td>TDNonScrollWindow</td>
<td>AIPSTODRAW 17</td>
</tr>
<tr>
<td>TDPaint</td>
<td>AIPSTODRAW 18</td>
</tr>
<tr>
<td>TDRectangle</td>
<td>AIPSTODRAW 19</td>
</tr>
<tr>
<td>TDRectangle</td>
<td>AIPSTODRAW 20</td>
</tr>
<tr>
<td>TDSlowWindow</td>
<td>AIPSTODRAW 21</td>
</tr>
<tr>
<td>TDText</td>
<td>AIPSTODRAW 22</td>
</tr>
<tr>
<td>TDTTWindow</td>
<td>AIPSTODRAW 23</td>
</tr>
<tr>
<td>TDWindow</td>
<td>AIPSTODRAW 24</td>
</tr>
<tr>
<td>TDEDisplayAtom</td>
<td>AIPSTODRAW 25</td>
</tr>
<tr>
<td>TDEDisplayComposite</td>
<td>AIPSTODRAW 26</td>
</tr>
<tr>
<td>TemplateCGCoversTemplateCGP</td>
<td>AIPSDISPLAY 27</td>
</tr>
<tr>
<td>TemplateOutSpecifiesTemplateP</td>
<td>AIPSDISPLAY 28</td>
</tr>
<tr>
<td>TLAperture</td>
<td>AIPSWINDOW 10</td>
</tr>
<tr>
<td>TLColumnTable</td>
<td>AIPSTABLE 26</td>
</tr>
<tr>
<td>TLDWindow</td>
<td>AIPSDISPLAY 29</td>
</tr>
<tr>
<td>TLDdisplayAtom</td>
<td>AIPSREALIZATION 16</td>
</tr>
<tr>
<td>TLDdisplayItem</td>
<td>AIPSWINDOW 19</td>
</tr>
<tr>
<td>TLDdisplaySurface</td>
<td>AIPSWINDOW 20</td>
</tr>
<tr>
<td>TLPPlaneSurface</td>
<td>AIPSWINDOW 21</td>
</tr>
<tr>
<td>TLRectangle</td>
<td>AIPSREALIZATION 17</td>
</tr>
<tr>
<td>TLRowTable</td>
<td>AIPSTABLE 27</td>
</tr>
<tr>
<td>TLWindow</td>
<td>AIPSWINDOW 11</td>
</tr>
<tr>
<td>TMAperture</td>
<td>AIPSWINDOW 12</td>
</tr>
<tr>
<td>TMDisplaySurface</td>
<td>AIPSWINDOW 22</td>
</tr>
<tr>
<td>TMFastWindow</td>
<td>AIPSWINDOW 13</td>
</tr>
<tr>
<td>TMNonScrollWindow</td>
<td>AIPSWINDOW 14</td>
</tr>
<tr>
<td>TMRectangle</td>
<td>AIPSREALIZATION 19</td>
</tr>
<tr>
<td>TMRegionNumber</td>
<td>AIPSWINDOW 23</td>
</tr>
<tr>
<td>TMSlowWindow</td>
<td>AIPSWINDOW 15</td>
</tr>
<tr>
<td>TMTTWindow</td>
<td>AIPSWINDOW 16</td>
</tr>
<tr>
<td>TransformPoints</td>
<td>AIPSTODRAW 38</td>
</tr>
<tr>
<td>TransformSize</td>
<td>AIPSTODRAW 39</td>
</tr>
<tr>
<td>TSAperture</td>
<td>AIPSWINDOW 17</td>
</tr>
<tr>
<td>TSNonScrollWindow</td>
<td>AIPSWINDOW 18</td>
</tr>
<tr>
<td>TSPlaneSurface</td>
<td>AIPSWINDOW 24</td>
</tr>
<tr>
<td>TSRectangle</td>
<td>AIPSREALIZATION 20</td>
</tr>
<tr>
<td>TSSRectangularRegion</td>
<td>AIPSWINDOW 25</td>
</tr>
<tr>
<td>TSWindow</td>
<td>AIPSWINDOW 19</td>
</tr>
<tr>
<td>TT2DLinearTransFix</td>
<td>AIPSCS 14</td>
</tr>
<tr>
<td>TT2DLinearTransform</td>
<td>AIPSCS 15</td>
</tr>
<tr>
<td>TTRansFix</td>
<td>AIPSCS 16</td>
</tr>
<tr>
<td>TTranslation</td>
<td>AIPSCS 17</td>
</tr>
<tr>
<td>TTXPosition</td>
<td>AIPSDOMAIN 16</td>
</tr>
<tr>
<td>TTXWorldPosition</td>
<td>AIPSGEOGRAPHY 9</td>
</tr>
</tbody>
</table>
2. SOURCE FILE: AIPS

(FILECREATED "8-Nov-80 22:49:37" <NEWAIPS>AIPS..3 686
changes to: AIPSUTILITYFNS SendMessage AIPS_VARS AIPSCOMS
previous date: "7-Nov-80 21:02:56" <NEWAIPS>AIPS..4)

(PRETTYCOMPRINT AIPSCOMS)

(RPAQQ AIPSCOMS ((FILES (COMPILED FROM NEWAIPS)
AIPSCS AIPSDEVICES AIPSDISPLAY AIPSDOMAIN
AIPSGEOGRAPHY AIPSMAP AIPSNAVAL AIPSNTDS
AIPSREALIZATION AIPSTABLE AIPSTEMP AIPSTODRAW
AIPSTS AIPSUTILITY AIPSWINDOW)))

(FILELOAD (COMPILED FROM NEWAIPS)
AIPSCS AIPSDEVICES AIPSDISPLAY AIPSDOMAIN AIPSGEOGRAPHY AIPSMAP
AIPSNAVAL AIPSNTDS AIPSREALIZATION AIPSTABLE AIPSTEMP AIPSTODRAW
AIPSTS AIPSUTILITY AIPSWINDOW)

(DECLARE: DONTCOPY
(FILEMAP (NIL)))
STOP

Source File: AIPS
3. SOURCE FILE: AIPSCS

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init2DCOORDINATESYSTEM</td>
<td>1</td>
</tr>
<tr>
<td>Init2DLINEARTRANSFIX</td>
<td>2</td>
</tr>
<tr>
<td>Init2DLINEARTRANSFORM</td>
<td>3</td>
</tr>
<tr>
<td>InitCARTESIANSYSTEM</td>
<td>4</td>
</tr>
<tr>
<td>InitCOORDINATESYSTEM</td>
<td>5</td>
</tr>
<tr>
<td>InitCoordinateSystemConcepts</td>
<td>6</td>
</tr>
<tr>
<td>InitINTEGERSYSTEM</td>
<td>7</td>
</tr>
<tr>
<td>InitLINEARTRANSFORM</td>
<td>8</td>
</tr>
<tr>
<td>InitMAPPING</td>
<td>9</td>
</tr>
<tr>
<td>InitORTHOGONALSYSTEM</td>
<td>10</td>
</tr>
<tr>
<td>InitTRANSFIX</td>
<td>11</td>
</tr>
<tr>
<td>InitTRANSLATION</td>
<td>12</td>
</tr>
<tr>
<td>InitVIEWSYSTEM</td>
<td>13</td>
</tr>
<tr>
<td>TT2DLinearTransFix</td>
<td>14</td>
</tr>
<tr>
<td>TT2DLinearTransform</td>
<td>15</td>
</tr>
<tr>
<td>TTTransFix</td>
<td>16</td>
</tr>
<tr>
<td>TTTTranslation</td>
<td>17</td>
</tr>
</tbody>
</table>
(FILECREATED "3-Jul-80 17:03:28" <NEWAIPS>AIPSCS..10 9183
changes to: AIPSCSCOMS
previous date: "3-Jul-80 13:50:27" <NEWAIPS>AIPSCS..9)

(PRETTYCOMPRINT AIPSCSCOMS)

(RPAQQ AIPSCSCOMS ((FNS * AIPSCSINITFNS)
(FNS * AIPSCSTOTRANSFORMFNS)
(ADDVARS (CKLONEFIES AIPSCS))))

(RPAQQ AIPSCSINITFNS (Init2DCOORDINATESYSTEM Init2DLINEARTRANSFIX
Init2DLINEARTRANSFORM
InitCARTESIANSYSTEM
InitCOORDINATESYSTEM
InitCoordinateSystemConcepts
InitINTEGERSYSTEM
InitLINEARTRANSFORM
InitMAPPING
InitORTHOGONALSYSTEM
InitTRANSFIX InitTRANSLATION
InitVIEWSYSTEM))

(DEFINEQ

(Init2DCOORDINATESYSTEM
[LAMBDA NIL (* Edited by J.Gibbons on 2-Jul-80.)

[concept 2DCOORDINATESYSTEM
(specializes COORDINATESYSTEM)
[roleset NIL
(mods Dimensionality@COORDINATESYSTEM)
(vr 2)])])

(Init2DLINEARTRANSFIX
[LAMBDA NIL (* Edited by J.Gibbons on 3-Jul-80.)

(* Two dimensional, linear transforms from cartesian coordinate systems to integral cartesian coordinate systems)

[concept 2DLINEARTRANSFIX

Source File: AIPSCS 12
(specializes 2DLINEARTRANSFORM)
roleset NIL
(mods To@2DLINEARTRANSFORM)
(vr VIEWSYSTEM)]
(itags (ToTransform TT2DLinearTransFix))])

(Init2DLINEARTRANSFORM
[LAMBDA NIL
(* Edited by J.Gibbons on 3-Jul-80.)
[concept 2DLINEARTRANSFORM
(specializes LINEARTRANSFORM)
[roleset NIL
(mods From@LINEARTRANSFORM)
(vr CARTESIANSYSTEM)]
[roleset NIL
(mods To@LINEARTRANSFORM)
(vr CARTESIANSYSTEM)]
(itags (ToTransform TT2DLinearTransform))])

(InitCARTESIANSYSTEM
[LAMBDA NIL
(* Edited by J.Gibbons on 2-Jul-80.)
[concept CARTESIANSYSTEM
(specializes 2DCOORDINATESYSTEM ORTHOGONALSYSTEM)])

(InitCOORDINATESYSTEM
[LAMBDA NIL
(* Edited by J.Gibbons on 2-Jul-80.)
[concept COORDINATESYSTEM
[roleset Name
(vr STRINGP)]

(* This Role must be explicitly included because it is necessary for distinguishing among Coordinate Systems who do not as yet have Entries and Exits. It is not desirable to use a Concept Name as other than an indexing mechanism at the implementational level.)
(* These are the transforms for leaving the Coordinate System.)

(* Transforms for entering the Coordinate System.)

(* Initializes those concepts having to do with Coordinate Systems and mappings between them. Some major changes from the previous paradigm occur among these concepts. For example, positions are now simply LISTP, and must be interpreted relative to some Coordinate System found in the context of the description referencing the position.)

(InitCOORDINATESYSTEM)
(InitINTEGERSYSTEM)
(Init2DCOORDINATESYSTEM)
(InitORTHOGONALSYS)
(InitCARTESIANSYSTEM)
(InitVIEWSYSTEM)
(InitMAPPING)
(InitLINEARTRANSFORM)
(Init2DLINEARTRANSFORM)
(Init2DLINEARTRANSFIX)
(InitTRANSLATION)
(InitTRANSFIX]}

(InitINTEGERSYSTEM
[LAMBDA NIL (* Edited by J.Gibbons on 2-Jul-80.)

[concept INTEGERSYSTEM (specializes COORDINATESYSTEM)
(* The purpose of this distinction is to signal that certain transformations must involve rounding. Bit maps are integer systems. *)

]]

(InitLINEARTRANSFORM
 [LAMBDA NIL
  (concept LINEARTRANSFORM
   (specializes MAPPING)
   [roleset NIL
    (mods From@MAPPING)
    (vr ORTHOGONALSYSTEM)]
   [roleset NIL
    (mods To@MAPPING)
    (vr ORTHOGONALSYSTEM)]
   [roleset Origin
    (vr LISTP)]
   [roleset UnitVector
    (vr LISTP)]))]

(InitMAPPING
 [LAMBDA NIL
  (concept MAPPING
   [roleset From
    (modality Obligatory)
    (number 1 NIL)
    (vr COORDINATESYSTEM)]
   [roleset To
    (modality Obligatory)
    (number 1 NIL)
    (vr COORDINATESYSTEM)]
  (* Should have an ITag whose referent is a function that can be applied to a list of positions and smash them with transformed coordinates. *)
]])
(InitORTHOGONALSYSTEM
[LAMBDA NIL

[concept ORTHOGONALSYSTEM

(* This Concept should have an SD that describes the
constraint of Orthogonality on the system's axes.)

(specializes COORDINATESYSTEM)])

[11]

(InitTRANSFIX
[LAMBDA NIL

[concept TRANSFIX
(specializes 2DLINEARTRANSFIX TRANSLATION)
(itags (ToTransform TTTransFix))]

[12]

(InitTRANSLATION
[LAMBDA NIL

[concept TRANSLATION

(specializes 2DLINEARTRANSFORM)
[roleset NIL
(mods UnitVector@2DLINEARTRANSFORM
(vr <l 1>))
(itags (ToTransform TTTranslation))]

[13]

(InitVIEWSYSTEM
[LAMBDA NIL

[concept VIEWSYSTEM

(* This kind of coordinate system can be used to describe
display regions.)

Source File: AIPSCS
(specializes INTEGERSYSTEM CARTESIANSYSTEM))}

(RPAQQ AIPSCSTOTRANSFORMFNS (TT2DLinearTransFix TT2DLinearTransform
TTTransFix TTTranslation))

(DEFINEQ

{14}

(TT2DLinearTransFix
[DLAMBDA ((transform IndividualConcept (SATISFIES (transform df
2DLINEARTRANSFIX)))
   (point (LISTP OF NUMBERP)))
   (CLISP: MIXED) (* Edited by J.Gibbons on
3-Jul-80.)

(* Performs a 2DLinearTransFixing -
a 2DLinearTransformation to integer coordinates -
on point according to transform. Returns the fixed, 2D
linearly transformed point.)

(PROG [(unitVector (OR (transform;UnitVector);1 '(1 1)))
   (origin (OR (transform;Origin);1 '(0 0)
   (RETURN '<(ROUND point;1*unitVector;1+origin;1)
     (ROUND point;2*unitVector;2+origin;2)
>))])

{15}

(TT2DLinearTransform
[DLAMBDA ((transform IndividualConcept (SATISFIES (transform df
2DLINEARTRANSFORM)))
   (point (LISTP OF NUMBERP)))
   (CLISP: MIXED) (* Edited by J.Gibbons on
3-Jul-80.)

(* Performs a 2DLinearTransformation on point according to
transform. Returns the 2D linearly transformed point.)

(PROG [(unitVector (OR (transform;UnitVector);1 '(1 1)))
   (origin (OR (transform;Origin);1 '(0 0)
   (RETURN '<point;1*unitVector;1+origin;1
     point;2*unitVector;2+origin;2>))])

{16}
((TTTransFix
 [DLAMBDAS ((transform IndividualConcept (SATISFIES (transform df `TRANSFIX)
 ))
 (point (LISTP OF NUMBERP)))
 (CLISP: MIXED) (* Edited by J.Gibbons on 3-Jul-80.)

 (* Performs a TransFixing -
 a Translation to integer coordinates -
 on point according to transform. Returns the fixed, translated
 point.)

 (PROG [[(origin (OR {transform;Origin})):l '(0 0)
 (RETURN <(ROUND point:l+origin:l)
 (ROUND point:2+origin:2)
 >)])

 )

 (TTTranslation
 [DLAMBDAS ((transform IndividualConcept (SATISFIES (transform df `TRANSLATION)))
 (point (LISTP OF NUMBERP)))
 (CLISP: MIXED) (* Edited by J.Gibbons on 3-Jul-80.)
 (* Performs a Translation on
 point according to transform.
 Returns the translated point.)

 (PROG [[(origin {transform;Origin})):l])
 (RETURN (if origin
 then <point:l+origin:l point:2+origin:2>
 else point))]
 )

 (ADDTOVAR CKLONEFILES AIPSCS)
 STOP

 Source File: AIPSCS
4. SOURCE FILE: AIPSDEVICES

InitCONRACSCREEN.....1
InitCTYWINDOW........2
InitDeviceConcepts...3
InitDevices..........4
InitHIDDENPLANE.....5
InitTURTLESSCREEN...6
(FILECREATED "19-Jan-81 01:23:51" <NEWAIPS>AIPSDEVICES..20 3933

changes to: InitDevices

previous date: "3-Jan-81 01:05:32" <NEWAIPS>AIPSDEVICES..19)

(PRETTYCOMPRINT AIPSDEVICESCOMS)

(RPAQQ AIPSDEVICESCOMS ((FNS * AIPSDEVICESEINITFNS)
   (GLOBALVARS TDRegion)
   (ADDVARS (CKLONEFILES AIPSDEVICES)))

(RPAQQ AIPSDEVICESEINITFNS (InitCONRACSCREEN InitCTYWINDOW
   InitDeviceConcepts InitDevices
   InitHIDDENPLANE
   InitTURTLESCREEN))

(DEFINEQ

{1}

(InitCONRACSCREEN
 [LAMBDA NIL
   (* Edited by J.Gibbons on 31-Dec-80.)
   (* This is the IC for the PLANESURFACE which maps to the BMG
   PLANESURFACE viewed on the CONRAC monitor.)

   (iconcept CONRACSCREEN (individuates PLANESURFACE VISIBLESURFACE)
      (irole Plane (vr 'CONRACSCREEN))
      (irole PlaneNumber (vr 1))
      (irole Boundary (vr (TMRectangle):l))
      (irole Background (vr 0))

   {2}

   (InitCTYWINDOW
    [LAMBDA NIL
     (* Edited by J.Gibbons on 3-Jan-81.)
     (* This TTYWindow supports the controlling terminal on the
     TurtleScreen.)

     (PROG (window)
        (window (SendMessage 'TTYWINDOW 'ToMake 'TURTLESCREEN):l)
        (KL2ChangeConceptName window 'CTYWINDOW)
        (SendMessage window 'ToSize ['TURTLESCREEN;&Height]:l

Source File: AIPSDEVICES 20
(InitDeviceConcepts [LAMBDA NIL (* Edited by J.Gibbons on 9-Dec-80.)
/* Initializes the concepts associated with the BMG devices. 
These include the three ICs representing the PLANESURFACES of bit mapped memory. They also include the IC for the initial TTYWINDOW on the TURTLE screen called CTYWINDOW. 
InitDevices must be called to realize these concepts.*/

(InitTURTLESCREEN) 
(InitCONFACSCREEN) 
(InitHIDDENPLANE])

(InitDevices [LAMBDA NIL (* Edited by J.Gibbons on 19-Jan-81.)
/* Initializes the BMG package and fleshes out the relevant PlanesSurfaces. Also creates and draws the Controlling TTYwindow on the TURTLESCREEN.*/

(if (ASKUSER NIL 'N "Can I merely Reset (as opposed to Init) BMG? " NIL T)='Y
  then (BMGReset)
  else (BMGInit))
(SendMessage 'CONRACSCREEN 'ToSize)
(SendMessage 'CONRACSCREEN 'ToLocate)
(SendMessage 'TURTLESCREEN 'ToSize)
(SendMessage 'TURTLESCREEN 'ToLocate)
(SendMessage 'HIDDENPLANE 'ToSize)
(SendMessage 'HIDDENPLANE 'ToLocate)
(InitCTYWINDOW)
(SendMessage 'CONRACSCREEN 'ToDraw)
(SendMessage 'TURTLESCREEN 'ToDraw)
(SendMessage 'HIDDENPLANE 'ToDraw)
(SendMessage 'CTYWINDOW 'ToDraw)
(InitHIDDENPLANE
 [LAMBDA NIL
 (* Edited by J.Gibbons on 31-Dec-80.)

 (* This is the IC for the PLANESURFACE which maps to the large
 BMG PLANE. This PLANE is not viewed on any monitor.)

 (iconcept HIDDENPLANE (individuates PLANESURFACE INVISIBLESURFACE)
  (irole Plane (vr "HIDDENPLANE"))
  (irole PlaneNumber (vr 2))
  (irole Boundary (vr (TMRectangle):1))
  (irole Background (vr 0))

 [6]

 (InitTURTLESCREEN
 [LAMBDA NIL
 (* Edited by J.Gibbons on 31-Dec-80.)

 (* This is the IC for the PLANESURFACE which maps to the BMG
 PLANE viewed on the TURTLE monitor.)

 (iconcept TURTLESCREEN (individuates PLANESURFACE VISIBLESURFACE)
  (irole Plane (vr "TURTLESCREEN"))
  (irole PlaneNumber (vr 0))
  (irole Boundary (vr (TMRectangle):1))
  (irole Background (vr 0))

 })

 (DECLARE: DOEVAL@COMPARE DONTCOPY

 (ADDTOVAR GLOBALVARS TDRegion)
 )

 (ADDTOVAR CKLONEFILES AIPSDEVICES)
 STOP

 Source File: AIPSDEVICES 22
5. SOURCE FILE: AIPSDISPLAY

<table>
<thead>
<tr>
<th>Function Description</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIPSGetSpecifiedMetaDescriptions</td>
<td>12</td>
</tr>
<tr>
<td>CollectTemplatesDescribingEntity</td>
<td>13</td>
</tr>
<tr>
<td>FindApplicableApplicationSlots</td>
<td>14</td>
</tr>
<tr>
<td>FindApplicationObjects</td>
<td>15</td>
</tr>
<tr>
<td>FindCostOfDisplay</td>
<td>16</td>
</tr>
<tr>
<td>FindCoveringIntendedApplication</td>
<td>17</td>
</tr>
<tr>
<td>FindTemplateObjects</td>
<td>18</td>
</tr>
<tr>
<td>FindTemplateSlots</td>
<td>19</td>
</tr>
<tr>
<td>GuageTemplateRGCoverageOfTemplate</td>
<td>20</td>
</tr>
<tr>
<td>InitDISPLAY</td>
<td>3</td>
</tr>
<tr>
<td>InitDISPLAYABSTRACTION</td>
<td>4</td>
</tr>
<tr>
<td>InitTEMPLATE</td>
<td>5</td>
</tr>
<tr>
<td>InitPRESENTATION</td>
<td>2</td>
</tr>
<tr>
<td>InitPresentationModel</td>
<td>1</td>
</tr>
<tr>
<td>InitTEMPLATE</td>
<td>6</td>
</tr>
<tr>
<td>InitTEXT</td>
<td>7</td>
</tr>
<tr>
<td>MakeColumnString</td>
<td>21</td>
</tr>
<tr>
<td>MakeDefaultTextString</td>
<td>8</td>
</tr>
<tr>
<td>MakeDisplayName</td>
<td>9</td>
</tr>
<tr>
<td>MakeTextHeight</td>
<td>10</td>
</tr>
<tr>
<td>MakeTextWidth</td>
<td>11</td>
</tr>
<tr>
<td>SelectDefaultDisplay</td>
<td>22</td>
</tr>
<tr>
<td>SelectDisplay</td>
<td>23</td>
</tr>
<tr>
<td>SelectFromAmongSufficientDisplays</td>
<td>24</td>
</tr>
<tr>
<td>SupplyConceptMetaIndicator</td>
<td>25</td>
</tr>
<tr>
<td>SupplyRoleMetaIndicator</td>
<td>26</td>
</tr>
<tr>
<td>TemplateCGCoversTemplateCGP</td>
<td>27</td>
</tr>
<tr>
<td>TemplateOutSpecifiesTemplateP</td>
<td>28</td>
</tr>
<tr>
<td>TLDisplay</td>
<td>29</td>
</tr>
</tbody>
</table>
changes to: InitDISPLAY
previous date: "19-Jan-81 01:02:42" <NEWAIPS>AIPSDISPLAY..46)

(RETRYCOMPRINT AIPSDISPLAYCOMS)

(RPAQQ AIPSDISPLAYCOMS ((FNS * AIPSDISPLAYINITFNS)
 (FNS * AIPSDISPLAYDERIVATIONFNS)
 (FNS * AIPSDISPLAYUTILITYFNS)
 (FNS * AIPSDISPLAYTOLOCATEFNS)
 (DECLARE: EVAL@COMPILEWHEN
 (NOT (BOUNDP (QUOTE BMGCOMS)))
 DONTCOPY
 (FILES <FONTWORK>BMGRECORDSANDVARS..0))
 (ADDVARS (CKLONEFILES AIPSDISPLAY))))

(RPAQQ AIPSDISPLAYINITFNS (InitPresentationModel InitPRESENTATION
 InitDISPLAY
 InitDISPLAYABSTRACTION
 InitTEMPLATE InitTEMPLATE
 InitTEXT))

(DEFINEQ {1}

(InitPresentationModel
 [LAMBDA NIL
  (* Edited by F.Zdybel on 22-Jul-80.)

  (* Initializes the Presentation Model for AIPS.
  These concepts describe the linkage between information being
  output and the structure of the output.)

  (InitPRESENTATION)
  (InitDISPLAY)
  (InitDISPLAYABSTRACTION)
  (InitTEMPLATE)
  (InitTEMPLATE)
  (InitTEXT)

  (* These initialization functions build SI-Net descriptions of
  the principal Display types, such as Map, Table, Menu, etc.)

Source File: AIPSDISPLAY 24
(InitTableConcepts)
(InitMapConcepts))

(InitPRESENTATION
 [LAMBDA NIL (* Edited by F.Zdybel on 4-Jul-80.)
 [concept PRESENTATION

 (* The top-level concept of the Presentation Model for AIPS. This description embraces all forms of information output, including synthesized natural language messages and synthesized speech, as well as graphic displays.)

 [roleset Application

 (* This Roleset is intended to express the binding of some particular (Individual) Presentation to a set of information which is to be displayed. Other Roles of the generic description of the Presentation may imply that further information may be involved in all Presentations of a particular type.)

 (number (0 NIL))
 (vr ITEMPLATE))
 [roleset Realization

 (* This Roleset characterizes the internal structure of the Presentation as a group of constituents, which might be anything. This Role is differentiated and modified by generic descriptions descendant from Presentation.)

 (number (0 NIL))
 (vr THING)]])

 (InitDISPLAY
 [LAMBDA NIL (* Edited by J.Gibbons on 19-Jan-81.)
 [concept DISPLAY

 Source File: AIPSDISPLAY
A Graphically realized Presentation.

(specializes PRESENTATION DISPLAYITEM)
[roleset NIL
  (mods Realization@PRESENTATION)
  (vr DISPLAYITEM)

(* A Display is either a Presentation or a Display Form.
Expansion of the description of any individual Display
terminates at the level of Display Forms
(the Viewing Organization and Realization Models of AIPS.))
]

[roleset Ground
  (modality Obligatory)
  (vr VIEWSURFACE)]

(* This Roleset describes the coordinate system and any
boundaries of the bit map surface on which the Display will be
realized.)

[roleset Name
  (vr STRINGP)
  (derivation (MakeDisplayName))
  (itags (ToDraw TDDisplay)
    (ToLocate TLDisplay))]]}

(InitDISPLAYABSTRACTION
 [LAMBDA NIL
  [concept DISPLAYABSTRACTION
   (specializes DISPLAYITEMABSTRACTION)
   [roleset IntendedApplication
    (number (1 NIL))
    (vr TEMPLATE)]

  (* This role is used in the abstraction of a Display to
indicate what kind of information the Display can depict.
In the case of MAP, for example, the role is filled at the
abstraction of map by two templates that indicate that a map
can be used to depict the locations of physical objects or the
boundaries of regions.)

Source File: AIPSDISPLAY
[roleset InformationRequirement
  (number (1 NIL))
  (vr TEMPLATE)]

(* Give a means of characterizing how much information will have to be known in order to realize a Display.)
]
]
]
}

(InitTEMPLATE
 [LAMBDA NIL (* Edited by F.Zdybel on 4-Jul-80.)
 [concept ITEMPLATE

(* A Template whose Concept Group Role restricts it to the description of slots inherited by Individual Concepts.)

(specializes TEMPLATE)
[roleset NIL
  (mods ConceptGroup@TEMPLATE)
  (vr ICONCEPT)])]]

(InitTEMPLATE
 [LAMBDA NIL (* Edited by F.Zdybel on 4-Jul-80.)
 [concept TEMPLATE

(* A Template is a meta-description that characterizes a set of inherited Rolesets in an SI-Net hierarchy. Its' function is to indicate a chunk of information, as for example in characterizing the Domain World extension that is involved in an individual of Presentation.)

[roleset ConceptGroup]

(* All Templates are individual descriptions. The satisfiers of the Concept Group Role meta-indicate
Concepts in the Domain World as a way of specifying sub-lattices of the Domain Model within which inheritance of the specified Rolesets (specified by the Role Group Role of Template) is to be considered as part of the set of information being described.

(number (0 NIL))
(vr CONCEPT)
[roleset RoleGroup]

(* The satisfiers of the Role Group Role meta-indicate Rolesets in the Domain World Model. Because of the inheritance of Rolesets, this is an intensional characterization of a set of inherited Roles. This characterization is limited in scope by the fillers of the Concept Group Role, which specify subsets of the SI-Net within which inheritance of the specified Roles is to be considered as generating the extension.)

(modality Obligatory)
(number (1 NIL))
(vr ROLE)]

(InitTEXT
[LAMBDA NIL
[concept TEXT

(* Instances of Text may be called on to display practically anything. There is therefore no IntendedApplication Template on TEXT's abstraction. Instead, TEXT is selected by default whenever no other Display with specified IntendedApplication can be discovered that discharges the purpose at hand.)

(specializes DISPLAY)
(abstraction [iconcept TEXTABSTR of DISPLAYABSTRACTION])
[roleset NIL
(mods Realization@DISPLAY)
(number 0)]

(* A kluge of sorts, TEXT itself has no Realization, since it is itself also a descendant of DISPLAYITEM.)

Source File: AIPSDISPLAY
[roleset String
  (modality Obligatory)
  (number 1)
  (vr STRINGP)
  (fillwhenmade)
  [derivation (MakeDefaultTextString (Prerequisites Application)
       (Arguments (CAR $(CONCEPT;Application} 11
       (roleset Font
         (vr SMALLP)
         (defaultfiller BMGDefaultRegionFontNumber))]
  (roleset LineSpacing
    (vr SMALLP)
    (defaultfiller BMGDefaultRegionLineSpacing))]
[roleset NIL
  (mods Height@DISPLAY)
  (derivation (MakeTextHeight (Prerequisites String Font LineSpacing)
        (Arguments [$(CONCEPT;String}:1
         [$(CONCEPT;Font}:1
         [$(CONCEPT;LineSpacing}:1))))
  [roleset NIL
    (mods Width@DISPLAY)
    (derivation (MakeTextWidth (Prerequisites String Font)
         (Arguments [$(CONCEPT;String}:1
                      [$(CONCEPT;Font}:1))))
    (itags (ToDraw TDText))]))
)
[RPAQQ AIPSDISPLAYDERIVATIONFNS (MakeDefaultTextString MakeDisplayName
MakeTextHeight
MakeTextWidth))

(DDEFINEQ

[8]

(MakeDefaultTextString
 [DLAMBDA ((application IndividualConcept (Satisfies (Application df 'ITEMPLATE
       (application df 'ITEMPLATE
       (FLENGTH
       [application;ConceptGroup]}=1
       (FLENGTH
       [application;RoleGroup]}=1)))
       (RETURNS (LST OF STRINGP)))]
       (* Edited by F.Zdybel on

[29]
Source File: AIPSDISPLAY
(* Makes the best possible effort to display application as a piece of text, incorporating Concept and Role Names where necessary. The RoleGroup of application is expected to consist of a single role metaindicator, the ConceptGroup of a single iconcept metaindicator. *)

(for roleFiller in (KLFindRoleValues (KLGetMetaDescribedEntity {application;ConceptGroup}:1 (KLGetNamedContext 'AIPSDOMAINMETA)) (KLGetMetaDescribedEntity {application;RoleGroup}:1 (KLGetNamedContext 'AIPSDOMAINMETA))

collect (if (KLConceptP roleFiller)
then (CAR (NLSETQ (SendMessage roleFiller 'ToTextify))
or (MKSTRING (OR {roleFiller;Name}:1 (KLGetConceptName roleFiller)))
elseif roleFiller = T
then "Yes"
elseif roleFiller = NIL
then "No"
else (MKSTRING roleFiller))
finally (RETURN <(if "$VAL":1 then "$VAL":1
else (MakeColumnString "$VAL")
>)))))

(MakeDisplayName
[DLAMBDAA ((RETURNS (LST OF STRINGP))]

("Edited by F.Zdybel on 17-Nov-80.")

("The last ditch effort to create a name for a Map is based on the name of the prototype map and a GENSYM.")

(PROG ((currentNumber (((KLGetData "$PROTOTYPE 'NumberCreatedSoFar"):1 or 0)+ 1))) (KLAttachDatum "$PROTOTYPE 'NumberCreatedSoFar currentNumber) (RETURN <(CONCAT (KLGetConceptName "$PROTOTYPE"
" " currentNumber)
>)))))
(MakeTextHeight
   [DLAMBDA ((string STRINGP)
         (fontNumber BMGFontNumber)
         (lineSpacing SMALLP))
   (* Edited by J.Gibbons on 12-Dec-80.)

   (* Calculates the Height of a TTYString in pixels according to
      the specified Font. NOTE: This also depends on the LineSpacing
      of the BMGRegion in which the TTYString will appear.
      We cannot handle this aspect currently and use the LineSpacing
      value of the current Graphics Region.)

   <(if string equal ""
      then 0
      else lineSpacing*(bind characterCode for character
            in (UNPACK string) eachtime characterCode_ (CHCON1 character)
            count characterCode=10 or characterCode=31)+(
            BMGDescribeFont fontNumber);LoadedFont\FontRecord:Font\Height)
   >])

(MakeTextWidth
   [DLAMBDA ((string STRINGP)
         (fontNumber BMGFontNumber))
   (* Edited by J.Gibbons on 15-Dec-80.)

   (* Calculates the maximum width of the given String in pixels,
      i.e. the width of its longest line, according to the specified
      font.)

   (bind characterCode currentWidth_0
         maximumWidth_0 for character in (UNPACK string)
         do (characterCode_(CHCON1 character))
         (if characterCode=13 or characterCode=31
            then maximumWidth_ (IMAX currentWidth maximumWidth)
            currentWidth_0
         elseif characterCode="10
            then (currentWidth_ currentWidth+(GetFontCharacterDef
                  fontNumber characterCode)
                  :CharDef\Width))
         finally (RETURN <(IMAX currentWidth maximumWidth)
   )

31 Source File: AIPSDISPLAY
(RPAQQ AIPSDISPLAYUTILITYFNS (AIPSGetSpecifiedMetaDescriptions
  CollectTemplatesDescribingEntity
  FindApplicableApplicationSlots
  FindApplicationObjects
  FindCostOfDisplay
  FindCoveringIntendedApplication
  FindTemplateObjects
  FindTemplateSlots
  GuageTemplateRGCoverageOfTemplate
  MakeColumnString
  SelectDefaultDisplay
  SelectDisplay
  SelectFromAmongSufficientDisplays
  SupplyConceptMetaIndicator
  SupplyRoleMetaIndicator
  TemplateCGCoversTemplateCGP
  TemplateOutSpecifiesTemplateP))

(DEFINEQ

131

(AIPSGetSpecifiedMetaDescriptions
  [DLAMBDA ((describedEntity MetaDescribableEntity)
    (metaDescRestriction Concept)
    (RETURNS (LST OF IndividualConcept))
    (* Edited by F.Zdybel on 19-Jun-80.)
    (* Finds all metadescsions of describedEntity that are eq
to or subCs of metaDescRestriction)

    (for metaDescription in (KLGetMetaDescriptions describedEntity)
      when (metaDescription=metaDescRestriction)
      or (metaDescription df metaDescRestriction)
      collect metaDescription)))

{13}

(CollectTemplatesDescribingEntity
  [DLAMBDA ((domainEntity IndividualConcept)
    (templateList (LST OF IndividualConcept) (SATISFIES
      (HomogenousConceptListP templateList `TEMPLATE)))
    (RETURNS (LST OF IndividualConcept) (SATISFIES
      (HomogenousConceptListP

12

Source File: AIPSDISPLAY

32
(for template in templateList
  when (for entityMetaDescription in {template;ConceptGroup}
    bind templateEntity eachtime templateEntity_
      KLGetMetaDescribedEntity entityMetaDescription
      (KLGetNamedContext 'AIPSDOMAINMETA))
    thereis domainEntity=templateEntity
      or (KLGenericConceptP templateEntity
        and domainEntity df templateEntity)
    and [for roleMetaDescription in {template;RoleGroup}
      thereis (KLFindRoleValues domainEntity
        (KLGetMetaDescribedEntity roleMetaDescription
        (KLGetNamedContext 'AIPSDOMAINMETA)
      collect template)])

(FindApplicableApplicationSlots
  [DLAMBDA ((domainEntity IndividualConcept)
    (templateList (LST OF IndividualConcept) (SATISFIES
      HomogenousConceptListP templateList `TEMPLATE)))
    (RETURNS (LST OF Role)))

(* Searches through the templates on templateList and collects together all of the roles metaindicated by templates that CG cover domainEntity.*)

(for template in templateList
  when (for templateObject in (FindTemplateObjects template)
    thereis (if (KLIndividualConceptP templateObject)
      then domainEntity=templateObject
      else domainEntity df templateObject))
  join (FindTemplateSlots template) finally (RETURN
    (INTERSECTION $$VAL $$VAL))))}}

Source File: AIPSDISPLAY
(FindApplicationObjects
[DLAMBDA ((templateList (LST OF IndividualConcept) (SATISFIES
HomogenousConceptListP templateList 'TEMPLATE)))
(RETURNS (LST OF Concept)))
/* Edited by F.Zdybel on 5-Aug-80. */
/* Goes through templateList and constructs a list of all the
domain entities indicated by the ConceptGroup fillers of the
templates. */
(for template in templateList join (FindTemplateObjects template)
finally (RETURN (INTERSECTION $$VAL $$VAL))))

(FindCostOfDisplay
[DLAMBDA ((display GenericConcept (SATISFIES (display df 'DISPLAY)))
(proposedApplication IndividualConcept (SATISFIES
(proposedApplication
df 'ITEMPLATE)))
(RETURNS NUMBERP))
/* Edited by F.Zdybel on 4-Jul-80. */
/* At the moment, the best measure we can get for the cost of
a display is the number of RoleGroup Role fillers of the
InformationRequirement that applies to the subjects
(ConceptGroup Role fillers) of proposedApplication.
The assumption is that the InformationRequirement templates on
a display's abstraction are disjoint with respect to subject. */
(FLENGTH {(for costTemplate in {(KLGetAbstraction display)
;InformationRequirement}
thereis (TemplateCGCoversTemplateCGP costTemplate
proposedApplication));RoleGroup})

(FindCoveringIntendedApplication
[DLAMBDA ((display GenericConcept (SATISFIES (KLIsConceptDescendantP

(proposedApplication IndividualConcept (SATISFIES (KLIsConceptDescendantP proposedApplication (VALUE 'TEMPLATE))))

(RETURNS (ONEOF (IndividualConcept (SATISFIES (KLIsConceptDescendantP VALUE 'TEMPLATE))
NIL))))

(* Finds the IntendedApplication template on display's abstraction that has to do with the subjects (ConceptGroup fillers) of proposedApplication. The assumption is that IntendedApplications on a display's abstraction are disjoint with respect to subject. *)

(for intendedApplication in {(KLGetAbstraction display)
  IntendedApplication
  thereis (TemplateCGCoversTemplateCGP intendedApplication proposedApplication)})

(FindTemplateObjects
[DLAMBD A ((template IndividualConcept (SATISFIES (KLIsConceptDescendantP template 'TEMPLATE))))
(RETURNS (LST OF Concept)))

(for objectDescriptor in {template;ConceptGroup}
collect (KLGetMetaDescribedEntity objectDescriptor (KLGetNamedContext 'AIPSDOMAINMETA) )))

(FindTemplateSlots
[DLAMBD A ((template IndividualConcept (SATISFIES (template df 'TEMPLATE))))
(RETURNS (LST OF Role)))

(for slotDescriptor in {template;RoleGroup}
collect (KLGetMetaDescribedEntity slotDescriptor (KLGetNamedContext 'AIPSDOMAINMETA) )))
(GuageTemplateRGCoverageOfTemplate
[DLAMBDA ((superTemplate IndividualConcept (SATISFIES (KLIsConceptDescendantP superTemplate 'TEMPLATE)))
  (subTemplate IndividualConcept (SATISFIES (KLIsConceptDescendantP subTemplate 'TEMPLATE)))
  (RETURNS FIXP))
  (* Edited by F.Zdybel on 30-Jul-80.)

(* Finds the number of RoleGroup fillers in superTemplate that
  metaindicate Roles that are ancestors of
  (or EQ to) the roles metaindicated by the RoleGroup fillers of
  subTemplate.)

(for superRoleMetaIndicator in {superTemplate;RoleGroup}
  bind (context _ (KLGetNamedContext 'AIPSDOMAINMETA))
  degreeOfCoverage_0
  superRole subRoles
  first subRoles_ (for subRoleMetaIndicator in {subTemplate;RoleGroup}
    collect (KLGetMetaDescribedEntity
      subRoleMetaIndicator
      context))
  eachtime superRole_ (KLGetMetaDescribedEntity superRoleMetaIndicator
    context)
  when (for subRole in subRoles thereis (KLIsRoleDescendantP subRole
    superRole))
  do degreeOfCoverage degreeOfCoverage+1
  finally (RETURN degreeOfCoverage))

(MakeColumnString
[LAMBDA (stringList) (* Edited by F.Zdybel on 9-Dec-80.)

(* Takes a list of strings and returns a single string which
  includes n-1 CRLF's among the n elements of the input list.)

(APPLY 'CONCAT (for fillerString in stringList::l
  bind (firstElt _ stringList:l)
  join "
" fillerString> finally (RETURN <firstElt ! $$VAL>))

Source File: AIPSDISPLAY
(SelectDefaultDisplay
  [DLAMBDA ((proposedApplication IndividualConcept (SATISFIES
    (proposedApplication df "ITEMPLATE")))
    (RETURNS GenericConcept (SATISFIES (VALUE df "DISPLAY")))
  )]
  (* Edited by F.Zdybel on 20-Jul-80.)

  (* Selects from either TABLE or TEXT depending on whether
    there is more than one RoleGroup filler in
    proposedApplication. This is a last ditch effort to find a way
    of displaying some information for which no tailored Display
    exists. Note that a Table or a Text can be arbitrarily
    combined into the containing Display even if neither TABLE or
    TEXT are value restrictions on that Display's Realization
    Role.)

  (if {proposedApplication;RoleGroup}:::1
    then 'TABLE
    else 'TEXT)]

  {23}

(SelectDefaultDisplay
  [DLAMBDA ((proposedApplication IndividualConcept (SATISFIES
    (KLIsConceptDescendantP proposedApplication
      'ITEMPLATE'))
    (possiblePrototypes (LST OF GenericConcept) (SATISFIES
      HomogenousConceptListP possiblePrototypes 'DISPLAY'))
    (RETURNS GenericConcept (SATISFIES (KLIsConceptDescendantP
      VALUE 'DISPLAY')))]
  (* Edited by F.Zdybel on 4-Aug-80.)

  (* Searches through possiblePrototypes to find the prototype
    MapItem that best suits applicationTemplate.
    At this point the general approach is to find the set of all
    displays with sufficient IntendedApplication to support
    proposedApplication (or to find the one display that most
    supports proposedApplication). If more than one display is
    applicable, we will (for the moment) take the most specific
    one. Eventually, the algorithm should go on to consider the
    cost of the candidate displays.)

  [for prototype in possiblePrototypes}
bind (maximumCoverage _ (FLENGTH {proposedApplication; RoleGroup!}))
  currentBestPrototype currentBestDegreeOfCoverage 0
  prototypeIntendedApplicationOfInterest
  candidateDegreeOfCoverage completelySufficientPrototypes
when prototypeIntendedApplicationOfInterest_
  FindCoveringIntendedApplication prototype proposedApplication
do (if candidateDegreeOfCoverage (GaugeTemplateRGCoverageOfTemplate
  prototypeIntendedApplicationOfInterest proposedApplication)
  = maximumCoverage
  then completelySufficientPrototypes
  < !! completelySufficientPrototypes prototype>
elseif candidateDegreeOfCoverage gt currentBestDegreeOfCoverage
  then currentBestPrototype prototype
  currentBestDegreeOfCoverage candidateDegreeOfCoverage
finally (RETURN (if completelySufficientPrototypes
  then (SelectFromAmongSufficientDisplays
     completelySufficientPrototypes
     proposedApplication)
else currentBestPrototype
  then currentBestPrototype
else (SelectDefaultDisplay proposedApplication)))

(SelectFromAmongSufficientDisplays
 [DLAMDBA ((alternativeDisplays (LST OF GenericConcept) (SATISFIES
  HomogenousConceptListP alternativeDisplays 'DISPLAY')))
  (proposedApplication IndividualConcept (SATISFIES
  (proposedApplication df 'ITEMPLATE)))
  (RETURNS GenericConcept (SATISFIES (VALUE df 'DISPLAY))))
  (* Edited by F. Zdybel on 18-Jul-80.)*

(* Tries to find the display of minimum information cost.
If several candidates have the same cost, tries to find the
most specialized.)*

[for prototypeDisplay in alternativeDisplays bind mostMinimalPrototypes
  (costOfMostMinimalPrototypes _ MAXFIX)
  eachtime costOfCandidate
  (FindCostOfDisplay prototypeDisplay proposedApplication)

(* We are assuming for the moment that the
InformationRequirements role is filled by at most one
template.)*

Source File: AIPSDISPLAY 38
do (if costOfCandidate lt costOfMostMinimalPrototypes
    then mostMinimalPrototypes_<
    prototypeDisplay>
  elseif costOfCandidate = costOfMostMinimalPrototypes
    then mostMinimalPrototypes_<!
      mostMinimalPrototypes
    prototypeDisplay>)
finally (RETURN (if ~mostMinimalPrototypes::l
    then mostMinimalPrototypes:1
    else (FindMostSpecializedConcept
      mostMinimalPrototypes]))

(SupplyConceptMetaIndicator
  [DLAMBDAG ((concept Concept)
    (contextName LITATOM)
    (RETURNS IndividualConcept))
  ]
  )

(* Edited by F.Zdybel on
11-Jul-80.)*

(* Finds or creates a metadescription for the given concept in
  either the given context or the first of the currently active
  contexts.)*

[OR (CAR (KLGetMetaDescriptions concept (if contextName
  then (KLGetNamedContext contextName)
  or
    (KLCreateContext contextName)
  else $$CONTEXTS::1))]

[iconcept of CONCEPT
  [contexts ((if contextName
    then (KLGetNamedContext contextName)
    or (KLCreateContext contextName)
  else $$CONTEXTS::1))
  (metadescibes (atomval concept))]]]

(SupplyRoleMetaIndicator
  [DLAMBDAD ((role Role)
    (contextName LITATOM)
    (RETURNS IndividualConcept))
  ]
  )

(* Edited by F.Zdybel on
39 Source File: AIPSDISPLAY
Either finds or creates a meta-description of the given role in either the given context or the first of the currently active contexts.

[OR (CAR (KLGetMetaDescriptions role (if contextName then (KLGetNamedContext contextName) or (KLCreateContext contextName) else $$CONTEXTS:1))))

[iconcept of ROLE
  contexts ((if contextName then (KLGetNamedContext contextName) or (KLCreateContext contextName) else $$CONTEXTS:1))
  (metadef (atomval role))]]]]]

(TemplateCGCoversTemplateCGP
[DLAMBDA ((superTemplate IndividualConcept (SATISFIES (superTemplate df 'TEMPLATE)))
  (subTemplate IndividualConcept (SATISFIES (subTemplate df 'TEMPLATE)))
  (RETURNS [ONEOF NIL (IndividualConcept (SATISFIES (VALUE df 'TEMPLATE))]
  (* Edited by F.Zdybel on 4-Jul-80.)

(PROG ((superObjects (FindTemplateObjects superTemplate))
  (subObjects (FindTemplateObjects subTemplate)))
  (RETURN (if (for subObject in subObjects
    always (for superObject in superObjects
      thereis subobject df superObject))
    then superTemplate
    else NIL))))]]]

(TemplateOutSpecifiesTemplateP
[DLAMBDA ((subTemplate IndividualConcept (SATISFIES (subTemplate df 'TEMPLATE)))
  (superTemplate IndividualConcept (SATISFIES (superTemplate df 'TEMPLATE)))
  (RETURNS [ONEOF NIL (IndividualConcept (SATISFIES (VALUE df 'TEMPLATE))]
  (* Edited by F.Zdybel on

Source File: AIPSDISPLAY 40
(* Tests to see whether the subTemplate names only objects that are descended from the objects named in the superTemplate, and that all of the slots named in the superTemplate are also included in the subTemplate. Put another way, this function checks to be sure that the ConceptGroup of superTemplate covers the ConceptGroup of subTemplate and the RoleGroup of subTemplate covers the RoleGroup of superTemplate. *)

(PROG ((superObjects (FindTemplateObjects superTemplate))
    (superSlots (FindTemplateSlots superTemplate))
    (subObjects (FindTemplateObjects subTemplate))
    (subSlots (FindTemplateSlots subTemplate)))
(RETURN (if (for subObject in subObjects
        always (for superObject in superObjects
            thereis subObject df superObject))
        and (for superSlot in superSlots
            always (for subSlot in subSlots
                thereis (KLIsRoleDescendantP
                    subSlot superSlot)))
        then subTemplate])))

(RPAQQ AIPSDISPLAYTOLOCATEFNS (TLDisplay))
(DEFINEQ

(TLDisplay
   [DLAMBDA ((displayDescr IndividualConcept (SATISFIES (displayDescr df 'DISPLAY)))
          (displayLoc (ONEOF NIL (LISTP OF FIXP (SATISFIES ~displayLoc::2))))
          (displayGround [ONEOF NIL (IndividualConcept (SATISFIES (displayGround df 'VIEWSURFACE)))]
              (* Edited by P. Zdybel on 16-Dec-80.))
          (* If given a ground, changes or establishes the Ground Role of displayDescr. If given a location, changes or establishes the Location Role of displayDescr. ))

[PROG (currentLocationRole currentGroundRole)
    (if displayLoc
        then (if currentLocationRole_ (KLFIndOneNamedInstanceRole
        "VIEWSURFACE"))
        else (if currentGroundRole (KLFIndOneNamedInstanceRole
        "DISPLAY"))
        else))

[Source File: AIPSDISPLAY]
displayDescr 'Location)
then (KLChangeRoleValue currentLocationRole
displayLoc)
else (KLSatisfyRole Location@DISPLAY displayDescr
<displayLoc>))

(if displayGround
then (if currentGroundRole_ (KLFindOneNamedInstanceRole
displayDescr 'Ground)
then (KLChangeRoleValue currentGroundRole
displayGround)
else (KLSatisfyRole Ground@DISPLAY displayDescr
<displayGround>))
)

(DECLARE: EVAL@COMPILE WHEN (NOT (BOUNDP (QUOTE BMGCOMS))) DONTCOPY
(FILESLoad <FONTWORK>BMGRECORDSANDVARS..0)
)

(ADDTOVAR CKLONFILES AIPSDISPLAY)
STOP
6. SOURCE FILE: AIPSDOMAIN

InitCommonDomainModel........1
InitDISTANCE..................2
InitDISTANCEUNIT..............3
InitINTERVAL..................4
InitINTERVALUNIT..............5
InitMETRIC.....................6
InitPHYSOBJECT................7
InitPLACE.....................8
InitPOSITION..................9
InitREGION....................10
InitSPEED.....................11
InitSPEEDUNIT................12
InitTIME.......................13
InitUNIT.......................14
InitVEHICLE....................15
TTXPosition...................16
(FILECREATED "10-Dec-80 23:41:31" <NEWAIPS>AIPSDOMAIN..5 8837
changes to: TTXPosition
previous date: "10-Dec-80 01:48:00" <NEWAIPS>AIPSDOMAIN..4)

(PRETTYCOMPRINT AIPSDOMAINCOMS)

(RPAQQ AIPSDOMAINCOMS ((FNS * AIPSDOMAININITFNS)
(FNS * AIPSDOMAINTOTEXTIFYFNS)
(ADDVARS (CKLONEFILES AIPSDOMAIN))))

(RPAQQ AIPSDOMAININITFNS (InitCommonDomainModel InitDISTANCE
InitDISTANCEUNIT
InitDISTANCEUNIT
InitINTERVALUNIT InitMETRIC
InitPHYSOBJECT InitPLACE
InitPLACE InitREGION
InitPOSITIVE InitSPEED InitSPEEDUNIT
InitTIME InitUNIT
InitVEHICLE))

(DEFINEQ

{1}

(InitCommonDomainModel
[LAMBDA NIL

(* Edited by F.Zdybel on
10-Jul-80.)

(* Sets up some of the very high level concepts of the domain
world, which are referenced by templates associated with
various Displays.)

(InitPHYSOBJECT)
(InitVEHICLE)
(InitPLACE)
(InitPOSITION)
(InitREGION)
(InitTIME)
(InitMETRIC)
(InitDISTANCE)
(InitINTERVAL)
(InitSPEED)
(InitUNIT)
(InitDISTANCEUNIT)
(InitINTERVALUNIT)
(InitSPEEDUNIT})

Source File: AIPSDOMAIN 44
(InitDISTANCE
 [LAMBDA NIL
  [concept DISTANCE
   (specializes METRIC)
   [roleset NIL
    (mods Unit@METRIC)
    (vr DISTANCEUNIT)]])]

(InitDISTANCEUNIT
 [LAMBDA NIL
  [concept DISTANCEUNIT
   (* Should establish a way of expressing equivalences among
distance units. For the moment merely establishes the
abbreviation for the unit.)
   (specializes UNIT)
   [roleset NIL
    (mods Subject@UNIT)
    (vr DISTANCE)]])]

(InitINTERVAL
 [LAMBDA NIL
  [concept INTERVAL
   (* The distance between two points in time.)
   (specializes METRIC)
   [roleset NIL
    (mods Unit@METRIC)
    (vr INTERVALUNIT)]])]
(InitINTERVALUNIT
 [LAMBDA NIL
  [concept INTERVALUNIT
   (specializes UNIT)
   [roleset NIL
    (mods Subject@UNIT)
    (vr INTERVAL)])])

(* Describes any unit used for measuring the passage of time.)

{6}

(InitMETRIC
 [LAMBDA NIL
  [concept METRIC
   (roleset Magnitude
    (vr NUMBERP))
   [roleset Unit
    (vr UNIT)])])

(* Describes measurement of something.)

{7}

(InitPHYSOBJECT
 [LAMBDA NIL
  [concept PHYSOBJECT
   (roleset Name
    (vr STRINGP))])

(* Edited by F.Zdybel on 9-Jul-80.)

(* For the moment, this is all that concerns us: objects are located somewhere in the domain world.)

{8}

(InitPLACE
 [LAMBDA NIL
  [concept PLACE
   (roleset Name
    (vr STRINGP))])

(* Edited by F.Zdybel on 9-Jul-80.)

{8}

Source File: AIPSDOMAIN 46
[concept PLACE]

(* This might be either a specific position or a region
(it also might be several other things like "over Paris"
"4:00 high" etc, but for the moment let us confine ourselves
...))

[roleset Constraint
  (vr THING)]

(* Might be a Point location (LISTP), or a curve, or the
boundary of a Region. For the moment we will eschew attempting
to characterize these things at this value restriction.)

[roleset Continuum
  (vr COORDINATESYSTEM)]

(* The constraint will be relative to this coordinate system.]]])

{9}

(InitPOSITION
  [LAMBDA NIL
    [concept POSITION
      (specializes PLACE)]

    [roleset Location
      (mods Constraint@PLACE)
      (vr LISTP)]

    (itags (ToTextify TTXPosition)))])

{10}

(InitREGION
  [LAMBDA NIL
    [concept REGION
      (specializes PLACE)]

    (concept POSITION
      (specializes PLACE)]

    [roleset Continuum
      (vr COORDINATESYSTEM)]

    (roleset Constraint
      (vr THING)]

    (* Might be a Point location (LISTP), or a curve, or the
boundary of a Region. For the moment we will eschew attempting
to characterize these things at this value restriction.)

    [roleset Constraint
      (vr THING)]

    (* Might be a Point location (LISTP), or a curve, or the
boundary of a Region. For the moment we will eschew attempting
to characterize these things at this value restriction.)

    [roleset Continuum
      (vr COORDINATESYSTEM)]

    (* The constraint will be relative to this coordinate system.]]])

{9}

(InitREGION
  [LAMBDA NIL
    [concept REGION
      (specializes PLACE)]

    (concept POSITION
      (specializes PLACE)]

    [roleset Continuum
      (vr COORDINATESYSTEM)]

    (roleset Constraint
      (vr THING)]

    (* Might be a Point location (LISTP), or a curve, or the
boundary of a Region. For the moment we will eschew attempting
to characterize these things at this value restriction.)

    [roleset Constraint
      (vr THING)]

    (* Might be a Point location (LISTP), or a curve, or the
boundary of a Region. For the moment we will eschew attempting
to characterize these things at this value restriction.)

    [roleset Continuum
      (vr COORDINATESYSTEM)]

    (* The constraint will be relative to this coordinate system.]]])

{9}
(* A Multi-purpose Concept. ITags on this concept should refer to functions that can decide whether a position is "inside" the region and can compute the distance to a boundary of the region from a given position. *)

[roleset Boundary
 (difs Constraint@PLACE)
 (vr CLOSEDCURVE)] (* A Region never has more than a single boundary. *)

[roleset Aperture
 (difs Constraint@PLACE)
 (number (0 NIL))
 (vr REGION)]

[roleset Height
 (vr NUMBERP)]
[roleset Width
 (vr NUMBERP)]

(InitSPEED
 [LAMBDA NIL (* Edited by F.Zdybel on 9-Jul-80.)

[concept SPEED

(specializes METRIC)
[roleset NIL
 (mods Unit@METRIC)
 (vr SPEEDUNIT)]]])

{11}

(InitSPEEDUNIT
 [LAMBDA NIL (* Edited by F.Zdybel on 9-Jul-80.)

[concept SPEEDUNIT

(* Links a unit for measuring speed to units for measuring distance and time. *)

(specializes UNIT)
[roleset DistanceUnit

Source File: AIPSDOMAIN
(* Like PLACE, TIME is a more or less specific constraint imposed on a CoordinateSystem. Eons and Instants are both Times. *)

[roleset Constraint
 (vr THING)]

(* This VR should become more elegant when we have time for elegance. Generally, the filler of this role is a NUMBERP.*)

[roleset Continuum
 (vr TIMESYSTEM)]

(* And there is (eventually) surely more than one TimeSystem (e.g.: Zulu time, GMT, Sidereal Time, Stardate, etc.)*)

(* A unit of measurement of some metric (e.g., speed, time, distance, temperature.)*)
(InitVEHICLE
 [LAMBDA NIL

 [concept VEHICLE

 (* A physical object capable of moving around under its own power and carrying something.)

 (specializes PHYSOBJECT)
 [roleset Medium
 (vr ATOM)]

 (* Expected to be something on the order of SubSurface, Air or Surface.)

 [roleset Course
 (vr NUMBERP)]

 (* Expected to be a number between 0 and 360, indicating the direction of motion of the Vehicle.)

 [roleset Speed
 (vr SPEED)]

 (* Gives the speed of the vehicle in terms of distance units per time unit.)

 ]])

 (RPAQQ AIPSDOMAINOTEXTIFYFNS (TTXPosition))
 (DEFINEQ

 (TTXPosition
 [LAMBDA (positionDescr)

 (* Edited by F.Zdybel on 13-Jul-80.)

 (* Attempts to construct a string which states the position described by positionDescr. At this level, this is merely a
(PROG ((coordinateList (CAR {positionDescr;Location})))
  (RETURN (APPLY 'CONCAT (for coordinate in coordinateList::1
    bind (firstElt coordinateList:1)
    join "", " coordinate"
    finally (RETURN"(" (MKSTRING firstElt)
      ! < !! $VAL ")" >>)))
  )
  )

(ADDTOVAR CKLONEFILES AIPSDOMAIN)
STOP
7. SOURCE FILE: AIPS GEOGRAPHY

InitGeographyModel... 2
InitGEOREGION....... 1
InitLANDFEATURE.... 3
InitLANDMASS........ 4
InitNATION.......... 5
InitWATERBODY...... 6
InitWATERFEATURE... 7
InitWORLDPOSITION.. 8
RptCoord........... 10
TTXWorldPosition... 9
(FILECREATED "11-Dec-80 00:01:03" <NEWAIPS>AIPSGEOGRAPHY .. 9 6330

changes to: AIPSGEOGRAPHYCOMS AIPSGEOGRAPHYUTILITYFNS TTXWorldPosition
previous date: "10-Dec-80 22:03:20" <NEWAIPS>AIPSGEOGRAPHY .. 7)

(PRETTYCOMPRINT AIPSGEOGRAPHYCOMS)

(RPAQQ AIPSGEOGRAPHYCOMS ((FNS * AIPSGEOGRAPHYINITFNS)
(FNS * AIPSGEOGRAPHYTOTEXTIFYFNS)
(FNS * AIPSGEOGRAPHYUTILITYFNS)
(ADDVARS (CKLONEFILES AIPSGEOGRAPHY)))))

(RPAQQ AIPSGEOGRAPHYINITFNS (InitGEOREGION InitGeographyModel
InitLANDFEATURE InitLANDMASS
InitNATION InitWATERBODY
InitWATERFEATURE
InitWORLDPOSITION))

(DEFINEQ

{1}

(InitGEOREGION
[LAMBDA NIL

(* Edited by P.Zdybel on
10-Dec-80.)

[concept GEOREGION

(* A region on surface of the earth that is modelled as having
a bounded area. This might describe cites, for example, but
not villages. It might describe some rivers, but not others.)

(specializes REGION GEOFEATURE)
[roleset Name
 (vr STRINGP)]
[roleset NIL
 (mods Continuum@REGION)
 (vr [iconcept WORLDCARTESSIAN])]

(* Notice that we are assuming for the moment a single world
coordinate system that is not spherical.)

[roleset Feature

Source File: AIPSGEOGRAPHY 54
(number (0 NIL))
(vr GEOFEATURE)

(* Notice that a feature may also be a region, so this allows us to cope with islands in the sea, cities on the land, etc.)

)(InitGeographyModel
 [LAMBDA NIL (* Edited by F.Zdybel on 10-Dec-80.)

(* This initialization function grows the descriptions for Geography Model. This portion of the Domain Model is generally necessary for complete description of a certain class of Maps which include outlines of land% masses, bodies of water, etc.)

(InitWORLDPOSITION)
(InitNATION)
(InitGEOREGION)
(InitWATERBODY)
(InitLANDMASS)
(InitWATERFEATURE)
(InitLANDFEATURE)]

)(InitLANDFEATURE
 [LAMBDA NIL (* Edited by F.Zdybel on 19-Jun-80.)

[concept LANDFEATURE
 (specializes GEOFEATURE)
 (subcs RIVER MOUNTAIN CITY PENINSULA ISTHMUS)]

)(InitLANDMASS
 [LAMBDA NIL (* Edited by F.Zdybel on 19-Jun-80.)

[concept LANDMASS
 (specializes GEOREGION GEOFEATURE)
 (subcs CONTINENT ISLAND CITY)
 [roleset Water
 (diffs Feature@GEOREGION)]]
(vr WATERBODY)
[roleset NIL
(diffs Feature@GEOREGION)
(vr LANDFEATURE)]]

(InitNATION
[LAMBDA NIL
(concept NATION

(* One of the top level concepts of the Geography Model. Further expansion of this Concept will make possible specified depiction of national attributes such as Capital, Seaport, IndustrialCenter, MilitaryInstallation, etc. The Geography model is actually a small Domain World in and of itself.)

[roleset Name
(vr STRINGP)]
[roleset Territory
(vr GEOREGION)]])

(InitWATERBODY
[LAMBDA NIL
(concept WATERBODY

(* One of two general classes of Geo Regions ...)

(specializes GEOREGION GEOFEATURE)
(subcs OCEAN SEA LAKE)
[roleset Land
(diffs Feature@GEOREGION)
(vr LANDMASS)

[roleset NIL
(diffs Feature@GEOREGION)
(vr WATERFEATURE)]

(* The Other of the two general classes ...)]

(* Note that a Water Body is allowed to contain land masses but not land features. Thus a city on an island in the sea is not necessarily retrieved as a Feature of the sea.)

])})
(InitWATERFEATURE
 [LAMBDA NIL
 (* Edited by F.Zdybel on 19-Jun-80.)
 [concept WATERFEATURE
 (specializes GEOFEATURE)
 (subcs REEF SEA STRAIGHT CHANNEL GULF)])]

(InitWORLDPOSITION
 [LAMBDA NIL
 (* Edited by F.Zdybel on 10-Dec-80.)
 [concept WORLDPOSITION

 (* A position expressed in world coordinates
 (hopefully soon to be actually a spherical system, but
 currently fudged as rectangular radians.))

 (specializes POSITION)
 [roleset NIL
 (mods Continuum@POSITION)
 (vr [iconcept WORLDCARTEESIAN of CARTESIANSYSTEM])
 (itags (ToTextify TTXWorldPosition))]])
)

(RPAQQ AIPSGEOGRAPHYTOTEXTIFYFNS (TTXWorldPosition))
(DEFINEQ

(TTXWorldPosition
 [LAMBDA (positionDescr)
 (* Edited by F.Zdybel on 11-Dec-80.)

 (* Takes spherical coordinates expressed in radians and outputs
 a string phrased in terms of Latitude and Longitude.)

 (PROG ((coordinateList (CAR [positionDescr;Location])))
  (RETURN (APPLY 'CONCAT < (RptCoord coordinateList:1 T)
 " " (RptCoord coordinateList:2)
 >))

57 Source File: AIPSGEOGRAPHY
(RPAQQ AIPS GEOGRAPHY UTILITYFNS (RptCoord))

(DEFINEQ

(RptCoord
[LAMBDA (radianNum latFlg)
(CLISP: FAST FLOATING)

(* Edited by F.Zdybel on 10-Dec-80.)

(* Function turns a latitude or longitude expressed as floating point radians into a list of the form degrees minutes seconds and direction, with appropriate descriptive labels for the fields.)

(PROG (degrees minutes seconds token)
(degrees_radianNum*180.0/3.141593)
(minutes_ (degrees_ (degrees_ (FIX degrees)))*60.0)
(seconds_ (FIX minutes_ (minutes_ (FIX minutes)))*6000.0)/100.0)
(token_ (If radianNum gt 0.0
    then (if latFlg
        then 'N.
        else 'E.)
    elseif latFlg
        then 'S.
        else 'W.))
(RETURN <(if latFlg
    then "Lat."
    else "Lng.")
(ABS degrees)
"d"
(ABS minutes)
"":"
(ABS seconds)
"s" token>))

)
8. SOURCE FILE: AIPSMAP

EstablishDomainEntityPositionInCS........................................ 23
FindDomainCoordinateSystem.................................................. 8
FindMapEntitiesFromApplication.............................................. 9
FindMapItemPrototypes....................................................... 21
FindMappingBetweenCSs.......................................................... 22
FindScaleUnitDistance............................................................ 10
InitMAP..................................................................................... 1
InitMapConcepts.......................................................................... 6
InitMAPGRID.............................................................................. 2
InitMAPITEM............................................................................... 3
InitMAPLEGEND.......................................................................... 4
InitMAPWITHBACKGROUND........................................................... 5
InitRECTANGULARMAP.................................................................. 7
MakeGroundForMapWithBackground............................................. 11
MakeLegendApplication............................................................... 12
MakeLinearMapTransformAndGroundFromEntities......................... 13
MakeLinearMapTransformFromGroundAndEntities......................... 14
MakeMapItemApplication............................................................. 24
MakeMapItemLabel................................................................. 15
MakeMapItems........................................................................... 16
MakeMapLabel.......................................................................... 17
MakeMapTransformForMapWithBackground................................... 18
MakeRectangularMapBorder......................................................... 19
MakeSimpleMapLegend............................................................... 20
MapPositionDescription ........................................................... 25
(* Edited by F. Zdybel on 29-Oct-80. *)

(concept MAP

(* Map is a major Presentation form. *)

(specializes DISPLAY)

(abstraction [iconcept MAPABSTR of DISPLAYABSTRACTION])

[roleset EntityList
  (vr LISTP)
  (derivation (FindMapEntitiesFromApplication (Prerequisites Application)
              (Arguments [$$CONCEPT;Application])))
  (fillwhenmade)]

(* The EntityList Role give a place to determine the set of objects with discrete physical locations that are to be included in the Map. This set is used to drive many of the other derivation procedures on Roles of MAP.
For example, the functions used to derive the size of the Ground for the Map know how to make it big enough to include things with discrete locations, but not how to make it big enough to include arbitrarily shaped areas (in the future, hopefully, this can be fixed so that regions can also be included in the EntityList.) The EntityList Role also provides a place to resolve coercions of several symbols...*)
into a single symbol for purposes of uncluttering (e.g.: three ships become a task group.))

[roleset NIL
  (mods Name@DISPLAY)
  (modality Obligatory)
  (fillwhenmade)]

(* This is the string that gets made into a Label for the Map. The name should be based on the largest georegion included in the Map (if this can be determined) or on the Template Role of the Map if it is simple enough to be summarized in a short string. At least we will see something here of the form "MAP #" <sequential Map number>.)

[roleset DomainCoordSys
  (vr 2DCOORDINATESYSTEM)
  [derivation (FindDomainCoordinateSystem
    (Prerequisites EntityList)
    (Arguments (CAR {$$CONCEPT;EntityList})
    (KLFindVRsOfRole (KLFindOneNamedGenericRole
      $$CONCEPT
      'DomainCoordSys)
    (fillwhenmade))]]

(* This is the domain world CoordinateSystem that is assumed to be shared by all of the items being depicted. It is not the Coordinate System of the depictions. Some of the depicted items may have to have their domain world locations re-expressed in terms of this coordinate system.)

[roleset MapItemTransform
  (modality Obligatory)
  (vr MAPPING)]
[roleset Label
  (diffs Realization@DISPLAY)
  (vr TEXT)
  [derivation (MakeMapLabel (Prerequisites Name Ground)
    (Arguments (CAR {$$CONCEPT;Name})
    (CAR {$$CONCEPT;Ground})
    (fillwhenmade))]
[roleset Border
  (diffs Realization@DISPLAY)
  (vr CLOSED CURVE)]
(* For some types of projections, the border of the Map is not necessarily a Rectangle, or even a Polygon.)

[roleset Legend
  (diffs Realization@DISPLAY)
  (vr MAPLEGEND)
  (derivation (MakeSimpleMapLegend (Prerequisites Name Item)
    (Arguments (CAR {$$CONCEPT;Name})
     {$$CONCEPT;Item}))))

(fillwhenmade)]

(* This may be one of the places where one desires to have a part of a Presentation viewed through one window while other parts of the Presentation are viewed through another (i.e. when scrolling over a Map with a Window one may wish that the Legend is always in view.))

[roleset ReferenceGrid
  (diffs Realization@DISPLAY)
  (vr MAPGRID)]
[roleset Item
  (modality Obligatory)
  (number (1 NIL))
  (diffs Realization@DISPLAY)
  (vr MAPITEM)
  [derivation (MakeMapItems (Prerequisites EntityList Application Ground)
    (Arguments {$$CONCEPT;Application}{CAR {$$CONCEPT;EntityList}}
     {CAR {$$CONCEPT;Ground}])
     (fillwhenmade)]
     (* Those things having discrete locations in the domain world.)

(itags (MinInterItemSpacing 10)
 (MinItemSpread 150))

(* These factors are the minimum spacing (in pixels) between any two items on a Map, and the minimum permissible spread (in pixels) in any dimension among all the items, respectively.)

(tags (DefaultPrototype "RECTANGULARMAP"))
[iconcept MAPABSTR
  (* The abstract individual representing the class of all maps.)

Source File: AIPSMAP
The first and obvious intended use of a Map is to show the location of something.

* A second intended use of a Map is to show the boundaries of regions.

(InitMAPGRID
[LAMBDA NIL
[concept MAPGRID
(specializes DISPLAY)
[roleset ReferenceLine
(diffs Realization@DISPLAY)
(modality Obligatory)
(number (4 NIL))
(vr CURVE)])
[roleset ReferenceLabel
(diffs Realization@DISPLAY)
(modality Obligatory)
(number (4 NIL))
(vr TEXT)])

(* Edited by F.Zdybel on 19-Jun-80. *)

Source File: AIPSMAP
While attached procedures know how to realize the particular types of reference grids, we

will restrict ourselves to displaying the discrete locations and names of physical objects.

Source File: AIPSMAP
(* This is the funny little bar with the colored stripes and the numbers under it. Note that this part of the Legend is sensitive to the current scale of the map, which may change due to Window-driven scaling. *)

(* This is the little correspondence Table between map symbols and explanatory text. *)

(* Probably says something brilliant like "Legend". *)
(InitMapWithBackground
[lambda nil

(concept mapWithBackground)

(* Any kind of map that already has one or more display forms
on the FixedComponent Role of its abstraction
(for geographic regions and reference grid) and hence has a
fixed DomainCoordSys and very particular value restrictions
for the MapItemTransform and Ground Roles.
For example, the value restriction of the MapItemTransform
Role will be a Generic 2DLinearTransform whose Origin and
UnitVector roles are filled with particular LISTP
(even though the From and To Roles remain to be filled at the
various individuals of that Generic Concept.)

(specializes map)

(* There is no point in having an abstraction for this
particular type of Map, because we will generally always be
dealing with its generic descendants.)

[roleset nil
 (mods ground@map)
 (derivation (makeGroundForMapWithBackground
 (arguments (klfindVrsOfRole (klfindOneNamedGenericRole
 $$CONCEPT 'Ground)):1)))
]

[roleset nil
 (mods mapItemTransform@map)
 (derivation (makeMapTransformForMapWithBackground
 (prerequisites ground)
 (arguments (klfindVrsOfRole (klfindOneNamedGenericRole
 $$CONCEPT 'MapItemTransform)):1
 $$CONCEPT;DomainCoordSys}:1
 $$CONCEPT;Ground}:1))]

[roleset nil
 (mods referenceGrid@map)
 (number (0 0)))]

{6}

(InitMapConcepts
[lambda nil

(* Edited by F.Zdybel on

Source File: AIPSMAP
(* Initializes that portion of the AIPS Presentation Model having to do with Maps.)

(InitMAP)
(InitRECTANGULARMAP)
(InitMAPITEM)
(InitMAPLEGEND)
(InitMAPGRID)
(InitMAPWITHBACKGROUND)

(InitRECTANGULARMAP
 [LAMBDA NIL (* Edited by F.Zdybel on 29-Oct-80.)

[concept RECTANGULARMAP

(* This is a flat rectangular map, as opposed to some kind of polar or conic projection with a funny shaped border and non-constant values for the screen distance equivalent to a given domain world distance.)

(specializes MAP)
[roleset NIL
 (mods DomainCoordSys@MAP)
 (vr CARTESIANSYSTEM)]

(* For the moment we will limit ourselves here to cartesian domain coordinate systems)

[roleset NIL
 (mods MapItemTransform@MAP)
 (vr 2DLINEARTRANSFORM)
 [derivation (MakeLinearMapTransformFromGroundAndEntities
 (Prerequisites DomainCoordSys Ground EntityList)
 (Arguments (CAR $$CONCEPT;EntityList))
 (CAR $$CONCEPT;Ground))
 (CAR $$CONCEPT;DomainCoordSys)))
 (MakeLinearMapTransformAndGroundFromEntities
 (Prerequisites DomainCoordSys EntityList)
 (Arguments (CAR $$CONCEPT;EntityList))
 (CAR $$CONCEPT;DomainCoordSys))))

[roleset ScaleUnit
(* The filler of this role should be selected according to the user's convenience. It is used in constructing the reference grid and the scale ikon of the legend. For the moment, the filler of this role is simply defaulted to NM at NTDSMAP. *)

[roleset ScaleDistance
  (vr NUMBERP)
  [derivation (FindScaleUnitDistance (Prerequisites DomainCoordSys Ground MapItemTransform)
    {Arguments (CAR {$$CONCEPT;MapItemTransform})
     (CAR {$$CONCEPT;DomainCoordSys})
     (CAR {$$CONCEPT;Ground;Continuum})}))]

(* This is the distance equivalent to the filler of the Scale Unit Role in the Coordinate System of the Presentation. *)

[roleset NIL
  (mods Border@MAP)
  (vr RECTANGLE)
  [derivation (MakeRectangularMapBorder (Prerequisites Ground)
    {Arguments (CAR {$$CONCEPT;Ground;Boundary;LowerLeft})
     (CAR {$$CONCEPT;Ground;Boundary;UpperRight})
     (CAR {$$CONCEPT;Ground}))
    {fillwhenmade})]

[roleset NIL
  (mods Ground@MAP)
  (derivation (Prerequisites MapItemTransform))
  (itags (MapMarginFactor .07))]

(* This factor is used to derive the amount by which the ground should be larger than the maximum spread in item X or Y coordinates. *)

]})

Source File: AIPSMAP
(RPAQQ AIPSMAPDERIVATIONFNS (FindDomainCoordinateSystem FindMapEntitiesFromApplication FindScaleUnitDistance MakeGroundForMapWithBackground MakeLegendApplication MakeLinearMapTransformAndGroundFromEntities MakeLinearMapTransformFromGroundAndEntities MakeMapItemLabel? MakeMapItems MakeMapLabel MakeMapTransformForMapWithBackground MakeRectangularMapBorder MakeSimpleMapLegend))

(defineq

(FindDomainCoordinateSystem
 [DLAMBDA ((domainEntityList (LST OF IndividualConcept))
                   (coordSysRestrictions (LST OF GenericConcept)) (SATISFIES
                   (HomogenousConceptListP coordSysRestrictions 'COORDINATESYSTEM))
                   (RETURNS (LST OF IndividualConcept) (SATISFIES (VALUE:1 if
                                      coordSysRestrictions:1)))))

(* Edited by F.Zdybel on 13-Nov-80.)

(* Looks through the input entity list and finds the coordinate system that is a descendant of all of coordSysRestrictions and is used by the majority of the entries. If no coordinate system can be found associated with any of the entities that satisfies all of coordSysRestrictions, something more complicated should be done to find a suitable domain coordinate system and establish the necessary mappings. For the moment, however, we will content ourselves with calling ERROR.)

(for alternativeEntry
   in (for domainEntity in domainEntityList bind domainCoordSys domainCoordSysEntry coordinateSystemAlternatives

       do

       (* Make an Alist to accumulate the different coordinate systems used by the various domain entities, where the value of each entry is the list of domain entities sharing the coordinate system.)

69 Source File: AIPSMAP
(if domainCoordSysEntry (ASSOC domainCoordSys_
    \{domainEntity;Position;Continuum\}:1
    coordinateSystemAlternatives)
    then <! domainCoordSysEntry domainEntity>
    else coordinateSystemAlternatives
    <! coordinateSystemAlternatives
    \{domainCoordSys domainEntity\}>
    finally (RETURN coordinateSystemAlternatives))
bind currentCandidate (currentCandidatePopularity_0)
alternative popularityOfAlternative
when popularityOfAlternative (FLENGTH alternativeEntry) gt
currentCandidatePopularity
and (for restriction in coordSysRestrictions
    first alternative alternativeEntry:1
    always alternative df restriction)
do

(* Find the most used domain coordinate system that satisfies
all of the constraints.)

(currentCandidate alternative)
(currentCandidatePopularity popularityOfAlternative)
finally (if currentCandidate
    then (RETURN <currentCandidate>)
    else (ERROR
    "unable to establish acceptable domain coordinate system:
    $$CONCEPT DERIVENOBREAKFLG)%%%%)))

(FindMapEntitiesFromApplication
[DLAMBDA ((mapApplicationList (LST OF IndividualConcept) (SATISFIES
    HomogenousConceptListP mapApplicationList `ITEMPLE
    (RETURNS (LST OF (LST OF IndividualConcept)))
    (* Edited by F.Zdybel on 13-Nov-80.)

(* Examines mapApplicationList (the Application Role fillers
of Some Map) and determines all of the discretely locatable
physical objects that are described by these templates.
This list is helpful for a number of derivation functions
attached to Roles of MAP.)

(for template in mapApplicationList
    join (for domainObjectMetaDescription in {template;ConceptGroup}

Source File: AIPSMAP 70
bind domainObjectDescription
when domainObjectDescription (KLGetMetaDescribedEntity
domainObjectMetaDescription
(KLGetNamedContext 'AIPS_DOMAINMETA))
and [domainObjectDescription; Position; Constraint]: 1
df 'LISTP
collect domainObjectDescription)
finally (RETURN <(INTERSECTION $$VAL $$VAL)
>))

(FindScaleUnitDistance
[DLAMBDA ((scaleUnit IndividualConcept (SATISFIES (scaleUnit df
`DISTANCEUNIT)))
(domainCoordSys IndividualConcept (SATISFIES
(domainCoordSys df
`COORDINATESYSTEM)))
(vsCoordSys IndividualConcept (SATISFIES (vsCoordSys df
`CARTESIANSYSTEM)))
(REturns (LST OF FLOATP)))

(* Edited by F.Zdybel on 13-Nov-80.)

(* Determines the distance represented by scaleUnit in
domainCoordSys in terms of an increment along an axis of
vsCoordSys. Otherwise, the result is returned as FLOATP.
Apparently we will require to associate with each domain
coordinate system an equivalence relationship between distance
units (e.g.: one NM in the Lat/Long system is one minute of
arc))

(ERROR "derivation function not implemented yet.
*"))

(MakeGroundForMapWithBackground
[DLAMBDA ((groundVR GenericConcept (SATISFIES (groundVR df `VIEWSURFACE)))
(REturns (LST OF IndividualConcept) (SATISFIES (VALUE:1 df
`VIEWSURFACE))))

(* Edited by F.Zdybel on 13-Nov-80.)

(* Creates an individual of groundVR, a Generic ViewSurface
with Boundary Role already restricted to a particular
ClosedCurve. Since this Value Description will already be
inherited by the new individual, there should be no need to
copy it onto an IRole. Assumes that the function that derives
the MapItemTransform will fill in the Exit role on the
Continuum of the Ground.)

\[
\textbf{< (APPLY* 'Make (KLGetConceptName groundVR)
'Continuum
(APPLY* 'Make (KLGetConceptName \{groundVR;Continuum\};1)))}
\]

\[
[12]
\]

\[
\textbf{(MakeLegendApplication}
[DLAMBDA ((itemRestrictionList (LST OF GenericConcept) (SATISFIES )
HomogenousConceptListP itemRestrictionList 'MAPITEM))
(mapItemList (LST OF IndividualConcept) (SATISFIES )
HomogenousConceptListP mapItemList 'MAPITEM))
(RETURNS (LST OF IndividualConcept) (SATISFIES VALUE:1 df
'ITEMPLATE))}
(* Edited by F.Zdybel on 13-Nov-80.)

(* Looks through the items in the Map and devises an
application role filler for the Legend of the Map.)

\[
\textbf{< (Make ITEMPLATE ConceptGroup}
(for itemType in (for itemRestriction in itemRestrictionList
join <itemRestriction
!(KLFindSubConcepts itemRestriction
>
finally (RETURN (INTERSECTION $$VAL $$VAL)))
when (for item in mapItemList thereis (KLZIsConceptDescendantP
item itemType))
join (for template in [{(KLGetAbstraction itemType)
;IntendedApplication}]
collect (SupplyConceptMetaIndicator template
'AIPSINTERNALMETA)))
RoleGroup
(SupplyRoleMetaIndicator ConceptGroup@TEMPLATE 'AIPSINTERNALMETA)
)
\]
\]

\[
[13]
\]

\[
\textbf{(MakeLinearMapTransformAndGroundFromEntities}
[DLAMBDA ((domainEntityList (LST OF IndividualConcept))}

Source File: AIPSMAP
(* Finds the minimum separation between map entities and selects a transform such that this minimum distance just maps onto some minimum desirable visual separation in terms of pixels. Thus, since window scaling is always greater than 1, this visual distance may become larger due to scaling transform associated with a Window onto the VS, but it will never become smaller. As a side effect, derives the Ground role of $\text{\$CONCEPT}$, and establishes the new transform as the Exit and Entry Role fillers of the domainCoordinateSystem and the Continuum of the new ground, respectively. *)

(for tail on (for entity in domainEntityList when Position@entity collect (CAR ((EstablishDomainEntityPositionInCS entity domainCoordinateSystem) ;Location)))
bind location locationXCoord locationYCoord minX_MAX.FLOAT
maxX_MIN.FLOAT
maxY_MIN.FLOAT
minXSpacing_MAX.FLOAT
maxXSpacing_MAX.FLOAT
minYSpacing_MAX.FLOAT
maxYSpacing_MAX.FLOAT
minSimpleDistance_MAX.FLOAT
minRealDistance scaleFactor mapGround itemsWidth itemsHeight
GroundWidth groundHeight [minItemSpread
(CAR (KFLFindIData $\text{\$CONCEPT}$ 'MinItemSpread]
[mapMarginFactor (CAR (KFLFindIData $\text{\$CONCEPT}$ 'MapMarginFactor)
xMargin yMargin xOrigin yOrigin transform

do (location tail::l)
(locationXCoord_location:XCord)
(locationYCoord_location:YCoord)
(minX_(MIN locationXCoord minX))
(maxX_(MAX locationXCoord maxX))
(minY_(MIN locationYCoord minY))
(maxY_(MAX locationYCoord maxY))
(for otherLocation in tail::l bind currentXSpacing
currentYSpacing
currentSimpleDistance

  do (currentXSpacing_(ABS locationXCoord-otherLocation:XCord))
  (currentYSpacing_(ABS locationYCoord-otherLocation:YCoord))
  (currentSimpleDistance currentXSpacing+currentYSpacing)
  (if currentSimpleDistance lt minSimpleDistance
)
then minSimpleDistance currentSimpleDistance
minXSpacing_currentXSpacing
minYSpacing_currentYSpacing)
finally (minRealDistance (SQRT minXSpacing^2+minYSpacing^2))
(scaleFactor_ (MAX (CAR (KLFindIData $CONCEPT ')
MinInterItemSpacing)))
/minRealDistance minItemSpread/(maxY-minY)
minItemSpread/(maxX-minX))
(itemsWidth_ (maxX-minX)*scaleFactor)
(itemsHeight_ (maxY-minY)*scaleFactor)
(groundWidth_itemsWidth+(2*(
xMargin_itemsWidth*mapMarginFactor)))
(groundHeight_itemsHeight+(2*(
yMargin_itemsHeight*mapMarginFactor)))
(mapGround_ (Make VIEWSURFACE Continuum
(Make CARTESIANSYSTEM)
Boundary
(Make RECTANGLE UpperRight
<groundWidth groundHeight> LowerLeft
<0.0 0.0>)

(xOrigin_xMargin-scaleFactor*minX)
(yOrigin_yMargin-scaleFactor*minY)
(transform (_Make 2DLINEARTRANSFORM From
domainCoordinateSystem To
(CAR [mapGround;Continuum])
Origin <xOrigin yOrigin> UnitVector
<scaleFactor scaleFactor>))

[irole Ground of (atomval $CONCEPT)
 (vr (atomval mapGround))]
[irole Exit of (atomval domainCoordinateSystem)
 (vr (atomval transform))]
[irole Entry of (CAR [mapGround;Continuum])
 (vr (atomval transform))]
(RETURN <transform>)

{14}

(MakeLinearMapTransformFromGroundAndEntities
[DLAMBDA ((domainEntityList (LST OF IndividualConcept) (SATISFIES
domainEntityList::l))
(mapGround IndividualConcept (SATISFIES mapGround df
'VIEWSURFACE))
(domainCoordSys IndividualConcept (SATISFIES domainCoordSys df
'CARTESIANSYSTEM))
(REturns (LST OF IndividualConcept) (SATISFIES VALUE:1 df
'2DLINEARTRANSFORM)))

(CLISP: MIXED (RECORD (XCoord YCoord)))

(* Edited by F.Zdybel on 13-Nov-80.)
(* Determines the appropriate transform to be used to map items from domainCoordinateSystem into the Map, given that the ViewSurface of the map has already been determined. Does this by looking for the maximum spread in x or y domain coordinates among the items and mapping this onto either the height of width (depending upon whether the max spread was along y or x) of the Ground of the Map, minus some reasonable amount, so that no item is exactly on the Boundary of the ViewSurface. As a side effect, the Continuum of the domainCoordinateSystem gets its Exit Role filled with the new created transform, ditto the Entry Role of the Continuum of the Ground.)

(for entity in domainEntityList
  bind temp currentEntityLocation minX MAX.FLOAT
  minY MIN.FLOAT
  maxX MIN.FLOAT
  maxY MIN.FLOAT
  ll ur groundHeight groundWidth
  [mapMarginFactor (CAR (KLFindlData $CONCEPT 'MapMarginFactor) adjustedGroundHeight adjustedGroundWidth xMargin yMargin scaleFactor xOrigin yOrigin transform
  when Position@entity
    do (currentEntityLocation_ (CAR {{EstablishDomainEntityPositionInCS entity domainCoordinateSystem) ;Location}}))
  (minX_(MIN temp currentEntityLocation:XCoord minX))
  (maxX (MAX temp maxX))
  (minY_ (MIN temp currentEntityLocation:YCoord minY))
  (maxY (MAX temp maxY))
  finally \[groundWidth_ (fetch XCoord of (ur_ (CAR {mapGround;Boundary;UpperRight})
                  -(fetch XCoord of (ll_ (CAR {mapGround;Boundary;LowerLeft})
                    (groundHeight_ ur:YCoord-ll:YCoord)
                    (adjustedGrou\[width groundWidth-(2(*(xMargin mapMarginFactor*groundWidth)))
                    (adjustedGroundHeight groundHeight-(2*(yMargin mapMarginFactor*groundHeight)))
                    scaleFactor_ (MIN adjustedGroundHeight/(maxY-minY)
                    adjustedGroundWidth/(maxX-minX))
                    (xOrigin_ ll:XCoord+xMargin-scaleFactor*minX)
                    (yOrigin_ ll:YCoord+yMargin-scaleFactor*minY)
                    (transform_ (Make 2DLINEARTRANSFORM From domainCoordSys To
                    (CAR {mapRound;Continuum})
                    Origin <xOrigin yOrigin> UnitVector <scaleFactor scaleFactor>))
    [irole Exit of (atomval domainCoordSys)
(vr (atomval transform))
[irole Entry of (CAR {mapGround;Continuum})
(vr (atomval transform))
(RETURN transform))}

(MakeMapItemLabel?
[DLAMBDA ((application IndividualConcept (SATISFIES application df
`ITEMPLE))
 (mapGround IndividualConcept (SATISFIES mapGround df
`VIEWSURFACE))
(RETURNS (LST OF IndividualConcept) (SATISFIES VALUE:1 df
`TEXT)))
 (* Edited by F.Zdybel on 13-Nov-80.)

(* Looks for a Name Role on the first object meta-indicated by
the ConceptGroup Role filler of application.
Creates an instance of TEXT based on what it finds.)

(PROG ((context (KLGetNamedContext 'AIPSDOMAINMETA))
 (targetRole Name@PHYSOBJECT)
 (domainEntity (FindTemplateObjects application):1))
 (if (for roleMetalndicator in {application;RoleGroup}
     thereis (KLIsRoleDescendantP (KLGetMetaDescribedEntity
       roleMetaIndicator context)
       targetRole)
     and (KLIFindRoleValues domainEntity targetRole)
     then (RETURN <(Make TEXT String {domainEntity;Namel:1 Ground
       mapGround)>
     )
     else (StopFill))))])

(MakeMapItems
[DLAMBDA ((mapApplication (LST OF IndividualConcept) (SATISFIES
    HomogenousConceptListP mapApplication `ITEMPLE))
    (domainEntityList (LST OF IndividualConcept))
    (mapGround IndividualConcept (SATISFIES (KLIsConceptDescendantP
      mapGround `VIEWSURFACE)))
    (RETURNS (LST OF IndividualConcept) (SATISFIES
      (HomogenousConceptListP
       VALUE `MAPITEM)))
    (* Edited by F.Zdybel on

Source File: AIPSMAP

Bolt Beranek and Newman Inc.  Report No. 4752

(15)

(16)
(* Goes through the entity list of the Map and determines what kind of Display to individuate for each, based on the value restrictions of the Item roles (For each item in the entity list determines the Application role filler implied by the Application role of the Map, then tries to find some descendant of one of the VR's on mapItemRestrictions that can fulfill the template for the item, or at least fulfill it to the greatest extent possible.))

(if (for domainEntity in domainEntityList bind mapItemGeneric
    mapItemGenericIAT
    mapItemGenericIASubjects
    first (mapItemGeneric (KLFindVRSOfRole $$ROLE):1)
    (mapItemGenericIAT_ {(KLGetAbstraction mapItemGeneric)
    ;IntendedApplication})
    (mapItemGenericIASubjects_ (FindApplicationObjects
    mapItemGenericIAT))
  when (for subject in mapItemGenericIASubjects
    thereis domainEntity df subject)
  and (SupplyConceptMetaIndicator domainEntity
    'AIPSDOMAINMETA)
    `memb {$$CONCEPT;Item;Application;ConceptGroup}
  collect (APPLY* 'Make (KLGetConceptName mapItemGeneric)
    'Application
    (MakeMapItemApplication domainEntity
    mapItemGenericIAT
    mapApplication)
    'Ground mapGround))
else (StopFill)))

{17}

(MakeMapLabel
 [DLAMBDA ((mapName STRINGP)
    (mapGround IndividualConcept (SATISFIES (mapGround df
    'VIEWSURFACE)))
    (RETURNS (LST OF IndividualConcept) (SATISFIES (VALUE:1 df
    'TEXT))))
  (* Edited by F.Zdybel on
  13-Nov-80.)

(* Creates an individual of TEXT. At some other subCs of MAP, the map label might be a shaded region with the instance of TEXT superimposed on it, but for the moment we will exercise the ToLocate procedures a bit by putting the label within the Map proper.)
(Make TEXT String mapName Ground mapGround)
>
(MakeMapTransformForMapWithBackground
[DLAMBDA ((transformVR GenericConcept (SATISFIES (transformVR df 'MAPPING)
))
 (domainCoordSys IndividualConcept (SATISFIES
 (domainCoordSys df 'COORDINATESYSTEM))
 (mapGround IndividualConcept (SATISFIES (mapGround df
 'VIEWSURFACE))
 (RETURNS (LST OF IndividualConcept) (SATISFIES (VALUE:1 df
 'MAPPING)))))
 (* Edited by F.Zdybel on
 13-Nov-80.)

(* Creates an individual of transformVR, a Generic MAPPING
 with Origin and UnitVector Roles already restricted to
 particular LISTP. Since these Value Descriptions will already
 be inherited by the new individual, there should be no need to
 copy them onto IRoles. As a side effect, fills the Exit Role
 of the DomainCoordSys and the Entry Role of the Continuum of
 the Ground with the new Transform.)

(PROG (mapTransform groundCoordSys)
 (mapTransform_ (APPLY* 'Make (KLGetConceptName transformVR)
 'From domainCoordSys 'To groundCoordSys_
 [mapGround;Continuum];1))
 [irole Exit of (atomval domainCoordSys)
 (vr (atomval mapTransform))]
 [irole Entry of (atomval groundCoordSys)
 (vr (atomval mapTransform))]
 (RETURN <mapTransform>))
]

(MakeRectangularMapBorder
 [DLAMBDA ((mapGroundLowerLeft (LST OF NUMBERP))
 (mapGroundUpperRight (LST OF NUMBERP))
 (mapGround IndividualConcept (SATISFIES mapGround df
 'VIEWSURFACE))
 (RETURNS (LST OF IndividualConcept) (SATISFIES VALUE:1 df
 'RECTANGLE))))
 (* Edited by F.Zdybel on
 13-Nov-80.)

MAP 78
(* Useful in drawing the border around a flat rectangular Map. Does this by limning the edge of the ViewSurface that the map is onto with an instance of RECTANGLE)

<(Make RECTANGLE LowerLeft mapGroundLowerLeft UpperRight mapGroundUpperRight Ground mapGround)>

(MakeSimpleMapLegend

[DLAMBDA ((mapName STRINGP)
   (mapItemList (LST OF IndividualConcept))
   (RETURNS (LST OF IndividualConcept) (SATISFIES (VALUE:1 df "MAPLEGEND"))))

(* Edited by F.Zdybel on 13-Nov-80.)

(* Makes the simplest possible kind of Legend (i.e.: merely a correspondence table between the generic display forms and the classes of domain world entities they denote.) Does this by driving the created MapLegend with an application that designates the ConceptGroup roles of each of all of the IntendedApplications found for each of the Generic Items used by the Map.)

<(Make MAPLEGEND Label (Make TEXT String (CONCAT "LEGEND for " mapName))
   Application
   (MakeLegendApplication (KLFindVRsOfRole (KLFindOneNamedGenericRole $$CONCEPT 'Item'):l mapItemList))>

(RPAQQ AIPSMAPUTILITYFNS (FindMapItemPrototypes FindMappingBetweenCSs EstablishDomainEntityPositionInCS MakeMapItemApplication MapPositionDescription))

 DEFINEQ

(FindMapItemPrototypes

[DLAMBDA ((map IndividualConcept (SATISFIES (map df 'MAP')))

(* Edited by F.Zdybel on 17-Jul-80.)

79 Source File: AIPSMAP
(Finds all the possible map item prototypes which apply
under the value restrictions of mapPrototype's Item Role.)

(for topLevelMapItemRestriction in (KLPfindVRsOfRole
KLPfindOneNamedGenericRole map 'Item)) join
	<topLevelMapItemRestriction

	!(KLPfindSubConcepts
topLevelMapItemRestriction)
>)}

(FindMappingBetweenCSs
[DLAMBDA ((coordSys1 IndividualConcept (SATISFIES (coordSys1 df
"COORDINATESYSTEM"))
(coordSys2 IndividualConcept (SATISFIES (coordSys2 df
"COORDINATESYSTEM"))

(REturns IndividualConcept (SATISFIES (VALUE df "MAPPING"))))
(* Edited by F.Zdybel on
2-Jul-80.)

(* Tries to find an existing transformation out of coordSys1
into coordSys2. Should be able to chain together and create
subsuming transforms in doing this, but for the moment simply
looks for an already existing transform.)

(for mapping in {coordSys1;Exit} thereis mapping

memb {coordSys2;Entry})))

(EstablishDomainEntityPositionInCS
[DLAMBDA ((domainEntity IndividualConcept)
(coordinateSystem IndividualConcept (SATISFIES
(coordinateSystem df
"COORDINATESYSTEM"))

(REturns IndividualConcept (SATISFIES (VALUE df "POSITION")))
(* Edited by F.Zdybel on
9-Jul-80.)

(* Insures that one of the Values of domainEntity's Position
Role is a position in the given coordinateSystem.
If necessary, establishes a new Position IRole for
domainEntity as a side-effect.)

(OR (for positionDescription in {domainEntity;Position}
thereis {positionDescription;Continuum}:1 = coordinateSystem
(for positionDescription in [domainEntity;Position]
bind transform mappedPosition thereis transform_
    FindMappingBetweenCSs
    {positionDescription;Continuum}:1
    coordinateSystem)
finally mappedPosition_ (MapPositionDescription
    positionDescription
    transform)
    [irole Position of (atomval domainEntity)
    (vr (atomval mappedPosition))]
    (RETURN mappedPosition))))

{24}

(MakeMapItemApplication
[DLAMBDAA ((domainEntity IndividualConcept)
    (genericItemIntendedApplication (LST OF IndividualConcept)
        (SATISFIES (HomogenousConceptListP
genericItemIntendedApplication `TEMPLATE))))
    (mapApplication (LST OF IndividualConcept)
        (SATISFIES HomogenousConceptListP mapApplication
        `TEMPLATE)))
    (RETURNS IndividualConcept (SATISFIES (KIsConceptDescendantP
VALUE `ITEMPLATE))))

(* Edited by F.Zdybel on 5-Aug-80.)

(* Constructs a single ITemplate which describes domainEntity and includes all of roles mentioned in mapApplication in connection with domainEntity that are covered by roles mentioned in genericItemTemplateList in connection with superCs of domainEntity.)

(Make ITEMPLATE ConceptGroup (SupplyConceptMetaIndicator domainEntity
    `AIPSDOMAINMETA)
RoleGroup
    (for role in (FindApplicableApplicationSlots domainEntity
        mapApplication)
bind (itemApplicationSlots _ (FindApplicableApplicationSlots
domainEntity
genericItemIntendedApplication))
when (for superRole in itemApplicationSlots
    thereis (KIsRoleDescendantP role superRole))
collect (SupplyRoleMetaIndicator role `AIPSDOMAINMETA))))

{25}
(MapPositionDescription
[DLAMBDA ((position IndividualConcept (SATISFIES (KLIsConceptDescendantP
  position "POSITION")))
  (transform IndividualConcept (SATISFIES (KLIsConceptDescendantP
    transform "MAPPING")))
  (RETURNS IndividualConcept (SATISFIES (KLIsConceptDescendantP
    VALUE "POSITION"))))
(CLISP: MIXED (RECORD (XCoord YCoord)))

(* Edited by F.Zdybel on 31-Jul-80.)

(* Transforms a location expressed in one coordinate system
  via the given transform into a location expressed in a
  different coordinate system. The actual transformation is
  carried out by the most specific ToTransform procedure
  associated with the mapping.)

(Make POSITION Location (APPLY* (CAR (KLFindlData transform
  ToTransform)))
  transform
  (CAR {position;Location}))
  Continuum
  (CAR {transform;To}))]
)

(ADDTOVAR CKLONEFILES AIPSMAP)
STOP
9. SOURCE FILE: AIPSNVAL

InitENEMYPLANE............1
InitENEMYSHIP.............2
InitENEMYSUB.............3
InitFRIENDLYPLANE.........4
InitFRIENDLYSHIP..........5
InitFRIENDLYSUB...........6
InitNavalDomainModel.....7
InitPLANE..................8
InitPLATFORM...............9
InitSHIP...................10
InitSUB....................11
InitUNKNOWNPLANE.........12
InitUNKNOWNSHIP..........13
InitUNKNOWNSUB...........14
(FILECREATED "17-Jul-80 02:14:13" <NEWAIPS>AIPSNAVAL..3 4370
changes to: InitENEMYSHIP InitENEMYSUB
previous date: "13-Jul-80 17:21:25" <NEWAIPS>AIPSNAVAL..2)

(PRETTYCOMPRINT AIPSNAVALCOMS)

(RPAQQ AIPSNAVALCOMS ((FNS * AIPSNAVALINITFNS)
(ADDVARS (CKLONEFILES AIPSNAVAL))))

(RPAQQ AIPSNAVALINITFNS (InitENEMYPLANE InitENEMYSHIP InitENEMYSUB
InitFRIENDLYPLANE InitFRIENDLYSHIP
InitFRIENDLYSUB
InitNavalDomainModel InitPLANE
InitPLATFORM InitSHIP InitSUB
InitUNKNOWNPLANE InitUNKNOWNSHIP
InitUNKNOWNSUB))

(DEFINEQ

[InitENEMYPLANE
 [LAMBDA NIL
 [concept ENEMYPLANE
 (specializes PLANE PLATFORM)
 [roleset NIL
 (mods Ownership@PLATFORM)
 (vr 'Enemy)]]])

[InitENEMYSHIP
[LAMBDA NIL (* Edited by F.Zdybel on 14-Jul-80.)

 [concept ENEMYSHIP
 (specializes SHIP PLATFORM)
 [roleset NIL
 (mods Ownership@PLATFORM)
 (vr 'Enemy)]]])

[InitENEMYSUB
[LAMBDA NIL (* Edited by F.Zdybel on

Source File: AIPSNAVAL 84
[concept ENEMYSUB
    (specializes SUB PLATFORM)
    [roleset NIL
     (mods Ownership@PLATFORM)
     (vr 'Enemy))]])

(InitFRIENDLYPLANE
 [LAMBDA NIL
  [concept FRIENDLYPLANE
   (specializes PLANE PLATFORM)
   [roleset NIL
    (mods Ownership@PLATFORM)
    (vr 'Friendly))]])

(InitFRIENDLYSHIP
 [LAMBDA NIL
  [concept FRIENDLYSHIP
   (specializes SHIP PLATFORM)
   [roleset NIL
    (mods Ownership@PLATFORM)
    (vr 'Friendly))]])

(InitFRIENDLYSUB
 [LAMBDA NIL
  [concept FRIENDLYSUB
   (specializes SUB PLATFORM)
   [roleset NIL
    (mods Ownership@PLATFORM)
    (vr 'Friendly))]])

(InitNavalDomainModel
 [LAMBDA NIL
  (InitPLATFORM)
  (InitSHIP)
  (InitSUB)
  (InitPLANE)

(* Edited by F.Zdybel on 9-Jul-80.)

(* Knowledge about the domain of naval tactical platforms.)
(InitFRIENDLYSHIP)
(InitFRIENDLYSUB)
(InitFRIENDLYPLANE)
(InitENEMYSHIP)
(InitENEMYSUB)
(InitENEMYPLANE)
(InitUNKNOWNSHIP)
(InitUNKNOWNSUB)
(InitUNKNOWNPLANE)

[8]

(InitPLANE
[LAMBDA NIL
[* Edited by F.Zdybel on
11-Jul-80.]

[concept PLANE
(specializes VEHICLE)
[roleset NIL
(mods Medium@VEHICLE)
(vr 'Air)])])

[9]

(InitPLATFORM
[LAMBDA NIL
[* Edited by F.Zdybel on
9-Jul-80.]

[concept PLATFORM

(* Describes a vehicle with weapons systems and sensor
systems, and an engagement status)

(specializes VEHICLE)
[roleset EngagementStatus
(vr ATOM)]

(* The filler of this role should be either T or NIL depending
on whether the Platform currently has targets engaged or is
itself engaged as a target.)

[roleset Ownership
(vr ATOM)]

(* Expected to be either
Friendly, Enemy or Unknown.)])

Source File: AIPSNAVAL
[10] (InitSHIP
   [LAMBDA NIL
    [concept SHIP
     (specializes VEHICLE)
     [roleset NIL
      (mods Medium@VEHICLE)
      (vr 'Surface)]]])

[11] (InitSUB
   [LAMBDA NIL
    [concept SUB
     (specializes VEHICLE)
     [roleset NIL
      (mods Medium@VEHICLE)
      (vr 'SubSurface)]]])

[12] (InitUNKNOWNPLANE
   [LAMBDA NIL
    [concept UNKNOWNPLANE
     (specializes PLANE PLATFORM)
     [roleset NIL
      (mods Ownership@PLATFORM)
      (vr 'Unknown)]]])

[13] (InitUNKNOWNSHIP
   [LAMBDA NIL
    [concept UNKNOWNSHIP
     (specializes SHIP PLATFORM)
     [roleset NIL
      (mods Ownership@PLATFORM)
      (vr 'Unknown)]]])

[14] (InitUNKNOWNSUB
[LAMBDA NIL
  [concept UNKNOWN
   (specializes SUB PLATFORM)
   [roleset NIL
    (mods Ownership@PLATFORM)
    (vr 'Unknown))]]]
)

(ADDTOVAR CKLONEFILES AIPSNAVAL)
STOP
10. SOURCE FILE: AIPSNTDS

InitENEMYNTDSPLANE..............1
InitENEMYNTDSSHIP...............2
InitENEMYNTDSSUB...............3
InitFRIENDLYNTDSPLANE..........4
InitFRIENDLYNTDSSHIP...........5
InitFRIENDLYNTDSSUB............6
InitNTDSConcepts.................7
InitNTDSENGAGEMENTMARK.........8
InitNTDSITEM....................9
InitNTDSMAP.....................10
InitNTDSVELOCITYLEADER.........11
InitUNKNOWNNTDSPLANE...........12
InitUNKNOWNNTDSSHIP.............13
InitUNKNOWNNTDSSUB..............14
MakeNTDSEngagementMark?........15
MakeNTDSGroupStrengthMark?.....16
MakeNTDSVelocityLeader?........17
(FILECREATED "13-Nov-80 02:21:26" <NEWAIPS>AIPSNTDS..26 31797)
changes to: MakeNTDSEngagementMark? MakeNTDSGroupStrengthMark?
MakeNTDSVelocityLeader?
previous date: "7-Aug-80 19:53:29" <NEWAIPS>AIPSNTDS..25)

(PRETTYCOMPRINT AIPSNTDSCOMS)

(RPAQQ AIPSNTDSCOMS ((FNS * AIPSNTDSINITFNS)
(FNS * AIPSNTDSDERIVATIONFNS)
(ADDVARS (CLKONEFILES AIPSNTDS))))

(RPAQQ AIPSNTDSINITFNS (InitENEMYNTDSPLANE InitENEMYNTDSSHIP
 InitFRIENDLYNTDSPLANE InitFRIENDLYNTDSSHIP
 InitFRIENDLYNTDSSUB InitNTDSConcepts
 InitNTDSENGAGEMENTMARK InitNTDSITEM InitNTDSMAP
 InitNTDSVELOCITYLEADER InitUNKNOWNNTDSPLANE
 InitUNKNOWNNTDSSHIP InitUNKNOWNNTDSSUB))

(DEFINEQ

(InitENEMYNTDSPLANE
 (LAMBDA NIL (*Edited
 by F.Zdybel on
 13-Jul-80.)
 [concept ENEMYNTDSPLANE
 (* The symbol used in an
 NTDS-type Map to denote an
 enemy airplane.)
 (specializes NTDSITEM)
 (abstraction [iconcept ENEMYNTDSPLANEABSTR of DISPLAYABSTRACTION]])
 [iconcept ENEMYNTDSPLANEABSTR
 [irole IntendedApplication
 [vr [iconcept ENEMYNTDSPLANEIAPL]]]
 [irole FixedComponent
 [vr (Make EDGESET VertexList '((-6 0)
 (0 6)
 (6 0))]
 [irole FixedComponent
 (vr (Make POINT Location '(0 0)))]])

Source File: AIPSNTDS
(Init ENEMYNTDSSHIP
[LAMBDA NIL
(* Edited by F.Zdybel on 13-Jul-80.))

[concept ENEMYNTDSSHIP]

(* The symbol used in an NTDS-type Map to denote a enemy surface ship.)

(specializes NTDSITEM)
(abstraction [iconcept ENEMYNTDSSHIPABSTR of DISPLAYABSTRACTION]))
[iconcept ENEMYNTDSSHIPABSTR
[irole IntendedApplication]
(vr [iconcept ENEMYNTDSSHIPIAPL1]])
[irole FixedComponent
 (vr (Make EDGESET VertexList '((-6 0)
(0 6)
(6 0)
(0 -6)
(-6 0))
 Location '(0 0))))
[irole FixedComponent
 (vr (Make POINT Location '(0 0))))]
[iconcept ENEMYNTDSSHIPIAPL1]

(* Indicates that ENEMYNTDSSHIP applies only to the depiction
of EnemyShips. Otherwise it is much like the
IntendedApplication template for NTDSITEM.)

(individuates TEMPLATE)
[irole ConceptGroup
 (vr (SupplyConceptMetaIndicator [concept ENEMYSHIP]
 'AIPSDOMAINMETA)])
[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Ownership of PLATFORM]
 'AIPSDOMAINMETA)])
[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Medium of VEHICLE]
 'AIPSDOMAINMETA)])
[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Position of PHYSOBJECT]
 'AIPSDOMAINMETA)])
[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset EngagementStatus of PLATFORM]
 'AIPSDOMAINMETA)])
[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Course of VEHICLE]
 'AIPSDOMAINMETA)])
[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Speed of VEHICLE]
 'AIPSDOMAINMETA)])
[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Name of PHYSOBJECT]
 'AIPSDOMAINMETA))])]

(InitENEMYNTDSSUB
[LAMBDA NIL
 (* Edited by F.Zdybel on
13-Jul-80.)
[concept ENEMYNTDSSUB

Source File: AIPSNTDS
92
*(The symbol used in an NTDS-type Map to denote an enemy submarine.)*

```plaintext
(specializes NTDSITEM)
(abstraction [iconcept ENEMYNTDSSUBABSTR of DISPLAYABSTRACTION])
[iconcept ENEMYNTDSSUBABSTR
 [irole IntendedApplication
  (vr [iconcept ENEMYNTDSSUBIAPL1])]
 [irole FixedComponent
  (vr (Make EDGESET VertexList '((-6 0)
    (-6 -6)
    (6 0)])
  (vr FixedComponent
   (vr Make POINT Location '(0 0))))]
 [iconcept ENEMYNTDSSUBIAPL1]

(* Indicates that ENEMYNTDSSUB should apply only to the depictions of EnemySubs. Otherwise, it is very like the IntendedApplication template for NTDSITEM.)*

(individuates TEMPLATE)
[irole ConceptGroup
  (vr (SupplyConceptMetaIndicator [concept ENEMYSUB] 'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Ownership of PLATFORM] 'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Medium of VEHICLE] 'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Position of PHYSOBJECT] 'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset EngagementStatus of PLATFORM] 'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Course of VEHICLE] 'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Speed of VEHICLE] 'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Name of PHYSOBJECT] 'AIPSDOMAINMETA))])
```
(InitFRIENDLYNTDSPLANE
 [LAMBDA NIL

 [concept FRIENDLYNTDSPLANE

 (* The symbol used in an
 NTDS-type Map to denote a
 friendly airplane.)

 (specializes NTDSITEM)
 (abstraction [iconcept FRIENDLYNTDSPLANEABSTR of DISPLAYABSTRACTION)])

 [iconcept FRIENDLYNTDSPLANEABSTR
 [irole IntendedApplication
  [vr [iconcept FRIENDLYNTDSPLANEIAPL1]])
 [irole FixedComponent
  [vr (Make EDGESET VertexList '((-6.0 0.0)
   (-5.706339 1.854102)
   (-4.854102 3.526712)
   (-3.526712 4.854102)
   (-1.854102 5.706339)
   (0.0 6.0)
   (1.854102 5.706339)
   (3.526712 4.854102)
   (4.854102 3.526712)
   (5.706339 1.854102)
   (6.0 0.0))]

 [irole FixedComponent
  [vr (Make POINT Location '(0 0))])]

 [iconcept FRIENDLYNTDSPLANEIAPL1

 (* Indicates that FRIENDLYNTDSPLANE should apply only to the
 depictions of FriendlyPlanes. Otherwise, it is very like the
 IntendedApplication template for NTDSITEM.)

 (individuates TEMPLATE)
 [irole ConceptGroup
  [vr (SupplyConceptMetaIndicator [concept FRIENDLYPLANE]
   'AIPSDOMAINMETA)])

 [irole RoleGroup
  [vr (SupplyRoleMetaIndicator [roleset Ownership of PLATFORM]
   'AIPSDOMAINMETA)])

 [irole RoleGroup
  [vr (SupplyRoleMetaIndicator [roleset Medium of VEHICLE]
   'AIPSDOMAINMETA)])

 [irole RoleGroup
  [vr (SupplyRoleMetaIndicator [roleset Position of PHYSOBJECT]
   'AIPSDOMAINMETA)])

 [irole RoleGroup
  [vr (SupplyRoleMetaIndicator [roleset EngagementStatus of PLATFORM]
   'AIPSDOMAINMETA)])

 [irole RoleGroup]
Report No. 4752

(vr (SupplyRoleMetaIndicator [roleset Course of VEHICLE] 'AIPSDOMAINMETA))

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Speed of VEHICLE] 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Name of PHYSOBJECT] 'AIPSDOMAINMETA))]]

(InitFRIENDLYNTDSSHIP
 [LAMBDA NIL
 (* Edited by F.Zdybel on 13-Jul-80.]
 [concept FRIENDLYNTDSSHIP]

(* The symbol used in an NTDS-type Map to denote a friendly surface ship.)

(specializes NTDSITEM)
(abstraction [iconcept FRIENDLYNTDSSHIPABSTR of DISPLAYABSTRACTION])
[iconcept FRIENDLYNTDSSHIPABSTR
 [irole IntendedApplication
 (vr [iconcept FRIENDLYNTDSSHIPIALPL]])
 [irole FixedComponent
 (vr (Make CIRCLE Radius 6 Location '(0 0)))]
 [irole FixedComponent
 (vr (Make POINT Location '(0 0)))]
 [iconcept FRIENDLYNTDSSHIPIALPL]

(* Indicates that FRIENDLYNTDSSHIP should apply only to the depictions of FriendlyShips. Otherwise, it is very like the IntendedApplication template for NTDSITEM.)

(individuates TEMPLATE)
[irole ConceptGroup
 (vr (SupplyConceptMetaIndicator [concept FRIENDLYSHIP] 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Ownership of PLATFORM] 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Medium of VEHICLE] 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Position of PHYSOBJECT]}

Source File: AIPSNTDS
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset EngagementStatus of PLATFORM] 'AIPSDOMAINMETA))]

[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Course of VEHICLE] 'AIPSDOMAINMETA))]

[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Speed of VEHICLE] 'AIPSDOMAINMETA))]

[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Name of PHYSOBJECT] 'AIPSDOMAINMETA))]

[6]

(InitFRIENDLYNTDSSUB
[LAMBDA NIL

[concept FRIENDLYNTDSSUB

(* The symbol used in an NTDS-type Map to denote a friendly submarine.)

(specializes NTDSITEM)
(abstraction [iconcept FRIENDLYNTDSSUBABSTR of DISPLAYABSTRACTION])

[iconcept FRIENDLYNTDSSUBABSTR
  [irole IntendedApplication
    (vr [iconcept FRIENDLYNTDSSUBIAPI])]
  [irole FixedComponent
    (vr (Make EDGESET VertexList '((-6.0 0.0)
                                    (-5.706339 -1.854102)
                                    (-4.854102 -3.526712)
                                    (-3.526712 -4.854102)
                                    (-1.854102 -5.706339)
                                    (0.0 -6.0)
                                    (1.854102 -5.706339)
                                    (3.526711 -4.854102)
                                    (4.854102 -3.526711)
                                    (5.706339 -1.854102)
                                    (6.0 0.0))]

  [irole FixedComponent
    (vr (Make POINT Location '(0 0)))]]

[iconcept FRIENDLYNTDSSUBIAPI]

(* Indicates that FRIENDLYNTDSSUB should apply only to the depictions of FriendlySubs. Otherwise, it is very like the IntendedApplication template for NTDSITEM.)

Source File: AIPSNTDS
(individuates TEMPLATE)
[irole ConceptGroup
  (vr (SupplyConceptMetaIndicator [concept FRIENDLYSUB]
    'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Ownership of PLATFORM]
    'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Medium of VEHICLE]
    'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Position of PHYSOBJECT]
    'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset EngagementStatus of PLATFORM]
    'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Course of VEHICLE]
    'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Speed of VEHICLE]
    'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Name of PHYSOBJECT]
    'AIPSDOMAINMETA))]

(InitNTDSConcepts
  [LAMBDA NIL
    (* Edited by F. Zdybel on
     14-Jul-80.)
    (* Initializes the displays
      used for NTDS-type tactical
      maps.)
    (InitNTDSMAP)
    (InitNTDSITEM)
    (InitFRIENDLYNTDSSHIP)
    (InitFRIENDLYNTDSPLANE)
    (InitFRIENDLYNTDSUB)
    (InitENEMYNTDSSHIP)
    (InitENEMYNTDSPLANE)
    (InitUNKNOWNNTDSPLANE)
    (InitUNKNOWNNTDSSHIP)
    (InitUNKNOWNNTDSSUB)
    (InitNTDSENGAGEMENTMARK)
    (InitNTDSVELOCITYLEADER)])

97 Source File: AIPSNTDS
(InitNTDSENGAGEMENTMARK
[ LAMBDA NIL
   (* Edited by F. Zdybel on 17-Jul-80. *)
   [concept NTDSENGAGEMENTMARK

   (* This is the little horizontal line through the locator dot in an NTDS symbol which indicates that the depicted platform has engaged or been engaged by some other platform. *)

   (specializes DISPLAY)
   (abstraction [iconcept NTDSENGAGEMENTMARKABSTR of DISPLAYABSTRACTION])
   [iconcept NTDSENGAGEMENTMARKABSTR
      [irole FixedComponent
         [vr (Make LINESEGMENT EndPoint '((-6 0)
            (6 0))]
      [irole IntendedApplication
         (vr [iconcept NTDSENGAGEMENTMARKIAPL1])]]
   [iconcept NTDSENGAGEMENTMARKIAPL1
      (individuates TEMPLATE)
      [irole ConceptGroup
         (vr [SupplyConceptMetaIndicator [concept PLATFORM]
            'AIPSDOMAINMETA]])
      [irole RoleGroup
         (vr [SupplyRoleMetaIndicator [roleset EngagementStatus of PLATFORM]
            'AIPSDOMAINMETA]])]
   [concept NARROWNTDSENGAGEMENTMARK

   (* Because the box on which the NTDS Symbols for unknown platforms are based is one unit narrower than the circle for friendlys or the lozenge for enemies, it is necessary to have a separate type of engagement mark for the unknowns. *)

   (specializes NTDSENGAGEMENTMARK)
   (abstraction [iconcept NARROWNTDSENGAGEMENTMARKABSTR of DISPLAYABSTRACTION])
   [iconcept NARROWNTDSENGAGEMENTMARKABSTR
      [irole FixedComponent
         [vr (Make LINESEGMENT EndPoint '((-5 0)
            (5 0))]
      [irole IntendedApplication
         (vr [iconcept NTDSENGAGEMENTMARKIAPL1])])])

[9]
(InitNTDSITEM  
[LAMBDA NIL  
[concept NTDSITEM  
(* A Map Item that is one of the NTDS symbols.)  
(specializes MAPITEM)  
(abstraction [iconcept NTDSITEMABSTR of DISPLAYABSTRACTION])  
[roleset NIL  
(mods Application@DISPLAY)  
(derivation (Consequents EngagementMark VelocityLeader  
GroupStrengthMarker))])}  
[roleset VelocityLeader  
(diffs Realization@DISPLAY)  
(vr NTDSVELOCITYLEADER)  
(derivation (MakeNTDSVelocityLeader? (Prerequisites Application  
Ground)  
(Arguments  
{$$CONCEPT;Application}:1  
{$$CONCEPT;Ground}:1)))]  
(* This is the vector line that extends from the center of the NTDSItem, the length and orientation of which denote the velocity of the vehicle.)  
[roleset EngagementMark  
(diffs Realization@DISPLAY)  
(vr NDSENSAGEMENTMARK)  
(derivation (MakeNDSEngagementMark? (Prerequisites Application  
Ground)  
(Arguments  
{$$CONCEPT;Application}:1  
{$$CONCEPT;Ground}:1)))]  
(* This is the line through the center of the NTDSItem, which indicates whether the denoted vehicle has engaged (or been engaged by) as the target of some other vehicle)  
[roleset GroupStrengthMarker  
(diffs Realization@DISPLAY)  
(vr NTDSSSTRENGTHMARK)  
(derivation (MakeNTDSGroupStrengthMark? (Prerequisites Application)  
(Arguments  
{$$CONCEPT;Application}))))]  
(* This is the little hashmark through the left side of the

Source File: AIPSNTDS
NTDSItem which indicates whether it denotes one, few, or many vehicles.)

(* This template is used to indicate the intended application not only of NTDSITEM, but also in many cases the intended application of generic descendants of NTDSITEM (e.g.: FRIENDLYNTDSSHIP, ENEMYNTDSSUB.) The possible purposes of these little Displays include depicting the ownership, location, velocity, name, and engagement status of various vehicles. Of course, these symbols also display the general type of the vehicle (e.g., Ship, Sub), but at the moment there is no RoleSet on VEHICLE to redundantly echo the category, thus there is no way to get a grip on that particular information with a Template. This is probably a deficiency in Template.)

(individuates TEMPLATE)
[irc role ConceptGroup
 (vr (SupplyConceptMetaIndicator [concept SHIP] 'AIPSDOMAINMETA))]
[irc role ConceptGroup
 (vr (SupplyConceptMetaIndicator [concept SUB] 'AIPSDOMAINMETA))]
[irc role ConceptGroup
 (vr (SupplyConceptMetaIndicator [concept PLANE] 'AIPSDOMAINMETA))]
[irc role RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Ownership of PLATFORM] 'AIPSDOMAINMETA))]
[irc role RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Medium of VEHICLE] 'AIPSDOMAINMETA))]
[irc role RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Position of PHYSOBJECT] 'AIPSDOMAINMETA))]
[irc role RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset EngagementStatus of PLATFORM] 'AIPSDOMAINMETA))]
[irc role RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Course of VEHICLE] 'AIPSDOMAINMETA))]
[irc role RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Speed of VEHICLE] 'AIPSDOMAINMETA))]
[irc role RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Name of PHYSOBJECT] 'AIPSDOMAINMETA))]

Source File: AIPSNTDS
(InitNTDSMAP
 [LAMBDA NIL
 [concept NTDSMAP

 (specializes RECTANGULARMAP)
 [roleset NIL
 (mods ScaleUnit@RECTANGULARMAP)
 (itags (DefaultFiller "NM"))
 ]
 [roleset FriendlyShipSym
 (diffs Item@MAP)
 (modality Obligatory)
 (number (0 NIL))
 (vr FRIENDLYNTDSSHIP))
 ]
 [roleset FriendlySubSym
 (diffs Item@MAP)
 (modality Obligatory)
 (number (0 NIL))
 (vr FRIENDLYNTDSSUB))
 ]
 [roleset FriendlyPlaneSym
 (diffs Item@MAP)
 (modality Obligatory)
 (number (0 NIL))
 (vr FRIENDLYNTDSPLANE))
 ]
 [roleset EnemyShipSym
 (diffs Item@MAP)
 (modality Obligatory)
 (number (0 NIL))
 (vr ENEMYNTDSSHIP))
 ]
 [roleset EnemySubSym
 (diffs Item@MAP)
 (modality Obligatory)
 (number (0 NIL))
 (vr ENEMYNTDSSUB))
 ]
 [roleset EnemyPlaneSym
 (diffs Item@MAP)
 (modality Obligatory)
 (number (0 NIL))
 (vr ENEMYNTDSPLANE))
 ]
 [roleset UnknownShipSym
 (diffs Item@MAP)
 (modality Obligatory)
 (number (0 NIL))
 (vr UNKNOWNNTDSSHIP))
]

(* Edited by F.Zdybel on 6-Aug-80. *)

(* Intended to describe the familiar type of NTDS situation display. *)
[roleset UnknownSubSym
  (diffs Item@MAP)
  (modality Obligatory)
  (number (0 NIL))
  (vr UNKNOWNNTDSSUB)]
[roleset UnknownPlaneSym
  (diffs Item@MAP)
  (modality Obligatory)
  (number (0 NIL))
  (vr UNKNOWNNTDSPLANE)])

{11}

(InitNTDSVELOCITYLEADER
 [LAMBDA NIL
  (* Edited by F.Zdybel on 6-Aug-80.)
]
[concept NTDSVELOCITYLEADER

(* This is the line leading out from the locator dot of an NTDS Symbol which gives the course and speed of the platform)

(specializes DISPLAY)
(abstraction [iconcept NTDSVELOCITYLEADERABSTR of DISPLAYABSTRACTION])
[roleset Leader
  (modality Obligatory)
  (mods Realization@DISPLAY)
  (vr LINESEGMENT)
  (itags (DefaultFiller (Make LINESEGMENT))))]
[iconcept NTDSVELOCITYLEADERABSTR
[irole IntendedApplication
  (vr [iconcept NTDSVELOCITYLEADERIAPL1])]
[iconcept NTDSVELOCITYLEADERIAPL1
(individuates TEMPLATE)
[irole ConceptGroup
  (vr (SupplyConceptMetaIndicator [concept PLATFORM]
       'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Course of VEHICLE]
       'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Speed of VEHICLE]
       'AIPSDOMAINMETA))]

{12}

(InitUNKNOWNNTDSPLANE
 [LAMBDA NIL
  (* Edited by F.Zdybel on

Source File: AIPSNNTDS
[concept UNKNOWNNTDSPLANE]

(* The symbol used in an NTDS-type Map to denote an unknown airplane.)

(specializes NTDSITEM)
(abstraction [iconcept UNKNOWNNTDSPLANEABSTR of DISPLAYABSTRACTION])

[roleset NIL
  (mods EngagementMark@NTDSITEM)
  (vr NARROWNTDSENGAGEMENTMARK))

[iconcept UNKNOWNNTDSPLANEABSTR
  [irole IntendedApplication
    (vr [iconcept UNKNOWNNTDSPLANEIAPL1])]
  [irole FixedComponent
    (vr (Make EDGESET VertexList '((-5 0)
         (-5 5)
         (5 5)
         (5 0)))]
  [irole FixedComponent
    (vr (Make POINT Location '(0 0)))]

[iconcept UNKNOWNNTDSPLANEIAPL1]

(* Indicates that UNKNOWNNTDSPLANE should apply only to the depictions of UnknownPlanes. Otherwise, it is very like the IntendedApplication template for NTDSITEM.)

(individuates TEMPLATE)
[irole ConceptGroup
  (vr (SupplyConceptMetaIndicator [concept UNKNOWNPLANE]
       'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Ownership of PLATFORM]
       'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Medium of VEHICLE]
       'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Position of PHYSOBJECT]
       'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset EngagementStatus of PLATFORM]
       'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Course of VEHICLE]
       'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Speed of VEHICLE]
       'AIPSDOMAINMETA))]
[irole RoleGroup

103   Source File: AIPSNTDS
Bolt Beranek and Newman Inc. Report No. 4752

(vr (SupplyRoleMetaIndicator [roleset Name of PHYSOBJECT] 'AIPSDOMAINMETA)))

[13]

(InitUNKNOWNNTDSHIPPING [LAMBDA NIL] (* Edited by F.Zdybel on 7-Aug-80.)

[concept UNKNOWNNTDSHIPPING]

(* Can be used in an NTDSMAP to depict unclassified Ships, or Ships that have been classified as Unknown (as indicated by their Ownership Role.))

(specializes NTDSITEM)
(abstraction [iconcept UNKNOWNNTDSHIPPINGABSTR of DISPLAYABSTRACTION]]
[roleset NIL]
(mods EngagementMark@NTDSITEM)
(vr NARROWNTDSENGAGEMENTMARK)])
[iconcept UNKNOWNNTDSHIPPINGABSTR]
[irole IntendedApplication
  (vr [iconcept UNKNOWNNTDSHIPPINGIAPL1])]
[irole FixedComponent
  (vr (Make EDGESET VertexList '((-5 -5)
                   (-5 5)
                   (5 -5)
                   (-5 -5))
         Location '(0 0))))
[irole FixedComponent
  (vr (Make POINT Location '(0 0))))]
[iconcept UNKNOWNNTDSHIPPINGIAPL1]

(* Indicates that UNKNOWNNTDSHIPPING should apply only to the depictions of UnknownShips. Otherwise, it is very like the IntendedApplication template for NTDSITEM)

(individuates TEMPLATE)
[irole ConceptGroup
  (vr (SupplyConceptMetaIndicator [concept UNKNOWNSHIP] 'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Ownership of PLATFORM] 'AIPSDOMAINMETA))]
[irole RoleGroup
  (vr (SupplyRoleMetaIndicator [roleset Medium of VEHICLE] 'AIPSDOMAINMETA))]

Source File: AIPSNTDS 104
Report No. 4752

Bolt Beranek and Newman Inc.

[irole RoleGroup (vr (SupplyRoleMetaIndicator [roleset Position of PHYSOBJECT] 'AIPSDOMAINMETA))]

[irole RoleGroup (vr (SupplyRoleMetaIndicator [roleset EngagementStatus of PLATFORM] 'AIPSDOMAINMETA))]

[irole RoleGroup (vr (SupplyRoleMetaIndicator [roleset Course of VEHICLE] 'AIPSDOMAINMETA))]

[irole RoleGroup (vr (SupplyRoleMetaIndicator [roleset Speed of VEHICLE] 'AIPSDOMAINMETA))]

[irole RoleGroup (vr (SupplyRoleMetaIndicator [roleset Name of PHYSOBJECT] 'AIPSDOMAINMETA))]]

{14}

(InitUNKNOWNNTDSUB [LAMBDA NIL]

[concept UNKNOWNNTDSUB (* Edited by F.Zdybel on 7-Aug-80.)

(* The symbol used in an NTDS-type Map to denote an unknown submarine.)

(specializes NTDSITEM)
(abstraction [iconcept UNKNOWNNTDSUBABSTR of DISPLAYABSTRACTION])
roleset NIL
(mods EngagementMark@NTDSITEM)
(vr NARROWNTDSENGAGEMENTMARK])

[iconcept UNKNOWNNTDSUBABSTR]
[irole IntendedApplication (vr [iconcept UNKNOWNNTDSUBIAPL1])]
[irole FixedComponent (vr (Make EDGESET VertexList '((-5 0) (-5 -5) (5 -5) (5 0))]
[irole FixedComponent (vr (Make POINT Location '0 0))]
[iconcept UNKNOWNNTDSUBIAPL1]

(* Indicates that UNKNOWNNTDSUB should apply only to the depictions of UnknownSubs. Otherwise, it is very like the IntendedApplication template for NTDSITEM.)

(individuates TEMPLATE)
[irole ConceptGroup
 (vr (SupplyConceptMetaIndicator [concept UNKNOWNSUB]
 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Ownership of PLATFORM]
 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Medium of VEHICLE]
 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Position of PHYSOBJECT]
 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset EngagementStatus of PLATFORM]
 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Course of VEHICLE]
 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Speed of VEHICLE]
 'AIPSDOMAINMETA))]

[irole RoleGroup
 (vr (SupplyRoleMetaIndicator [roleset Name of PHYSOBJECT]
 'AIPSDOMAINMETA))])

(RPAQQ AIPSNTDSDERIVATIONFNS (MakeNTDSEngagementMark?
 MakeNTDSGroupStrengthMark?
 MakeNTDSVelocityLeader?))

(DEFINEQ

(MakeNTDSEngagementMark?
 [DLAMBDA ((template IndividualConcept (SATISFIES template df `ITEMPLATE))
 (mapGround IndividualConcept (SATISFIES mapGround df
 `VIEWSURFACE))
 (RETURNS (LST OF IndividualConcept) (SATISFIES VALUE:l df
 `NTDSENGAGEMENTMARK)))
 (* Edited by F.Zdybel on 13-Nov-80.)

(* Looks through the templates in applicationList
 (the Application Role fillers at the NTDSItem) and at the
 EngagementStatus of the depicted vehicle and determines
 whether or not to create an engagement mark as part of the
 NTDSItem. If the role should not be filled, the derivation is
 aborted. Otherwise, an instance of NTDSENGAGEMENTMARK or
 NARROWNTDSENGAGEMENTMARK is returned.)

Source File: AIPSNTDS 106
(PROG ((vehicle ((FindTemplateObjects template):1))
  (targetRole (EngagementStatus@PLATFORM))
  (context (KLGetNamedContext 'AIPSDOMAINMETA))
  (if (for roleMetaIndicator in {template;RoleGroup}
        thereis (KLIsRoleDescendantP (KLGetMetaDescribedEntity
                                        roleMetaIndicator context)
                      targetRole))
    and (KLFindRoleValues vehicle targetRole):1~=NIL
    then (RETURN '<(APPLY* 'Make (KLGetConceptName (CAR
                                     '$SPROTOTYPE;EngagementMark))
                        'Application
                        (Make ITEMPLATE ConceptGroup
                         (SupplyConceptMetaIndicator
                          vehicle 'AIPSDOMAINMETA)
                         RoleGroup
                         (SupplyRoleMetaIndicator
                          targetRole
                          'AIPSDOMAINMETA))
                        'Ground mapGround)
    else (StopFill)))

{16}

(MakeNTDSGroupStrengthMark?
 [DLAMBDA ((itemApplications (LST OF IndividualConcept) (SATISFIES
    HomogenousConceptListP itemApplications 'ITEMPLATE))
    (RETURNS (LST OF IndividualConcept) (SATISFIES VALUE:1 df
             'DISPLAYITEM)))
    (* Edited by F.Zdybel on 13-Nov-80.)
    (StopDerivation))

{17}

(MakeNTDSVelocityLeader?
 [DLAMBDA ((template IndividualConcept (SATISFIES (template df
               'ITEMPLATE))
            (mapGround IndividualConcept (SATISFIES mapGround df
               'VIEWSURFACE))
            (RETURNS (LST OF IndividualConcept) (SATISFIES VALUE:1 df
                   'NTDSVELOCITYLEADER)))
            (* Edited by F.Zdybel Leader on 13-Nov-80.))

(* Looks through the templates in applicationList
the Application Role fillers at the NTDSItem) and determines
whether or not to create a velocity leader as part of the NTDSItem. If the templates do not specify the course and the speed of the object being depicted, then the derivation is aborted. Otherwise, an instance of LINESEGMENT is returned.

(PROG ((vehicle (FindTemplateObjects template);1)
  (speedRole Speed@VEHICLE)
  (courseRole Course@VEHICLE)
  (context (KLGetNamedContext 'AIPSDOMAINMETA))
  (if (for roleMetaIndicator in [template;RoleGroup]
      thereis (KLIsRoleDescendantP (KLGetMetaDescribedEntity
        roleMetaIndicator context)
        courseRole))
    and (for roleMetaIndicator in [template;RoleGroup]
      thereis (KLIsRoleDescendantP (KLGetMetaDescribedEntity
        roleMetaIndicator context)
        speedRole))
    and (KLFindRoleValues vehicle courseRole)
    and (KLFindRoleValues vehicle speedRole)
    then (RETURN <(Make NTDSVELOCITYLEADER Application
        (Make ITEMPLATE ConceptGroup
          (SupplyConceptMetaIndicator vehicle
            'AIPSDOMAINMETA)
          RoleGroup
          <$><(SupplyRoleMetaIndicator
            courseRole
            'AIPSDOMAINMETA)
          (SupplyRoleMetaIndicator
            speedRole
            'AIPSDOMAINMETA)
          >)
          Ground mapGround)
      >)
    else (StopFill))))})

(ADDTOVAR CKLONEFILES AIPSNNTDS)
STOP
11. SOURCE FILE: AIPSREALIZATION

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitCIRCLE</td>
<td>1</td>
</tr>
<tr>
<td>InitCLOSED CURVE</td>
<td>2</td>
</tr>
<tr>
<td>InitCURVE</td>
<td>3</td>
</tr>
<tr>
<td>InitDISPLAYATOM</td>
<td>4</td>
</tr>
<tr>
<td>InitDISPLAYCOMPOSITE</td>
<td>5</td>
</tr>
<tr>
<td>InitDISPLAYFORM</td>
<td>6</td>
</tr>
<tr>
<td>InitDisplayFormConcepts</td>
<td>7</td>
</tr>
<tr>
<td>InitEDGESET</td>
<td>8</td>
</tr>
<tr>
<td>InitELLIPSE</td>
<td>9</td>
</tr>
<tr>
<td>InitLINESEGMENT</td>
<td>10</td>
</tr>
<tr>
<td>InitPOINT</td>
<td>11</td>
</tr>
<tr>
<td>InitPOLYGON</td>
<td>12</td>
</tr>
<tr>
<td>InitRealizationModel</td>
<td>15</td>
</tr>
<tr>
<td>InitRECTANGLE</td>
<td>13</td>
</tr>
<tr>
<td>InitREGULARPOLYGON</td>
<td>14</td>
</tr>
<tr>
<td>MakeRectangle</td>
<td>18</td>
</tr>
<tr>
<td>TLDisplayAtom</td>
<td>16</td>
</tr>
<tr>
<td>TLRectangle</td>
<td>17</td>
</tr>
<tr>
<td>TMRectangle</td>
<td>19</td>
</tr>
<tr>
<td>TSRectangle</td>
<td>20</td>
</tr>
</tbody>
</table>
changes to: TLDisplayAtom

previous date: "13-Jan-81 22:59:22" <NEWAIPS>AIPSREALIZATION..27)

(PRETTYCOMPRINT AIPSREALIZATIONCOMS)

(RPAQQ AIPSREALIZATIONCOMS ((* The AIPS Realization Model resides here.
This contains the lowest-level description of drawable graphical entities as well as
the routines which know how to draw, erase, and otherwise deal with these.)*
(FNS * AIPSREALIZATIONINITFNS)
(FNS * AIPSREALIZATIONTOLOCATEFNS)
(ADDVARS (CKLONEFILES AIPSREALIZATION))
(FNS * AIPSREALIZATIONDERIVATIONFNS)
(FNS * AIPSREALIZATIONTOMAKEFNS)
(FNS * AIPSREALIZATIONTOSIZEFNS)))

[DECLARE: DONTEVAL@LOAD DONTCOPY

(* The AIPS Realization Model resides here.
This contains the lowest-level description of drawable graphical entities as well as the routines which know how to draw, erase, and otherwise deal with these.)
)

(RPAQQ AIPSREALIZATIONINITFNS (InitCIRCLE InitCLOSEDCURVE InitCURVE
InitDISPLAYATOM
InitDISPLAYCOMPOSITE
InitDISPLAYFORM
InitDisplayFormConcepts
InitEDGESET InitELLIPSE
InitLINESEGMENT InitPOINT
InitPOLYGON InitRECTANGLE
InitREGULARPOLYGON
InitRealizationModel))

(DEFINEQ

{1}

Source File: AIPSREALIZATION 110
(InitCIRCLE
 [LAMBDA NIL
   (* Edited by J.Gibbons on 8-Nov-80.)
   [concept CIRCLE
    (specializes ELLIPSE)
    [roleset Radius
     (mods SemiMinorAxis @ ELLIPSE)
     (mods SemiMajorAxis @ ELLIPSE)]
    (itags (ToDraw TDCircle))])]

{2}

(InitCLOSEDCURVE
 [LAMBDA NIL
   (* Edited by J.Gibbons on 31-Jul-80.)
   [concept CLOSEDCURVE
    (specializes CURVE)

    (* May differ from CURVE by the addition of tagged procedures
     for determining whether a given point is within the
     Approximation (which is expected to be an edgeset that closes
     on itself.))
   ]]

{3}

(InitCURVE
 [LAMBDA NIL
   (* Edited by J.Gibbons on 8-Jul-80.)
   [concept CURVE
    (* We cannot really handle curves so we approximate them
     with an Edgeset.)

    (specializes DISPLAYATOM)
    [roleset Approximation
     (vr EDGESET)]
    (itags (ToDraw TDCurve))])]

{4}

(InitDISPLAYATOM
 [LAMBDA NIL
   (* Edited by J.Gibbons on 13-Jan-81.)
   [concept DISPLAYATOM

   ]]

Source File: AIPSREALIZATION
(* A Display Atom is the most primitive type of Display Form. It has no constituents and is the final descriptive level before descent into LISP drawing procedures. *)

(specializes DISPLAYFORM)
(itags (ToErase TEDisplayAtom)
(ToLocate TLDisplayAtom))}

(* Edited by J.Gibbons on 13-Jan-81. *)

[concept DISPLAYCOMPOSITE
(specializes DISPLAYFORM)
[roleset Component
(modality Obligatory)
(number (1 NIL))
(vr DISPLAYFORM)]
[roleset CoordSys
(vr CARTESIANSYSTEM)]

(* The coordinate system by which the locations of the component Display Forms are interpreted. *)

(itags (ToDraw TDDisplayComposite)
(ToErase TEDisplayComposite))]

(* Edited by J.Gibbons on 13-Jan-81. *)

[concept DISPLAYFORM

(* Display Items come in two flavors: Displays (graphic Presentations) and Display Forms (graphic phenomena.))

(specializes DISPLAYITEM)])

Source File: AIPSREALIZATION
(InitDisplayFormConcepts
 [LAMBDA NIL (*Edited by J.Gibbons on 27-Oct-80.)
  (InitDISPLAYFORM)
  (InitDISPLAYATOM)
  (InitDISPLAYCOMPOSITE)
  (InitPOINT)
  (InitEDGESET)
  (InitCURVE)
  (InitLINESEGMENT)
  (InitCLOSEDCURVE)
  (InitELLIPSE)
  (InitPOLYGON)
  (InitCIRCLE)
  (InitREGULARPOLYGON)
  (InitRECTANGLE))

(InitEDGESET
 [LAMBDA NIL (*Edited by J.Gibbons on 8-Jul-80.)
  (concept EDGESET
   (specializes DISPLAYATOM)
   [roleset VertexList
    (modality Obligatory)
    (vr LISTP)]
   (itags (ToDraw TDEdgeSet)))]

(InitELLIPSE
 [LAMBDA NIL (*Edited by J.Gibbons on 13-Jan-81.)
  (concept ELLIPSE
   (specializes CLOSEDCURVE)
   [roleset Center
    (mods Location@CLOSEDCURVE)]
   [roleset SemiMinorAxis
    (diffs Size@CLOSEDCURVE)
    (modality Obligatory)]
   [roleset SemiMajorAxis
    (diffs Size@CLOSEDCURVE)
    (modality Obligatory)])

Source File: AIPSREALIZATION
(InitLINESEGMEMT
[LAMBDA NIL
[concept LINESEGMEMT
(specializes CURVE)
[roleset EndPoint
(mods Location@CURVE)
(number 2)]
[roleset Line
(vr LINE)
(derivation (DeriveLineEquationFromEndpoints (Prerequisites EndPoint)))]
(itags (ToDraw TDLineSegment))])

{10}

(InitPOINT
[LAMBDA NIL
[concept POINT
(specializes DISPLAYATOM)

(* The Location of a Point is given in a list of coordinates.)

(itags (ToDraw TDPoint))])

{11}

(InitPOLYGON
[LAMBDA NIL
[concept POLYGON
(specializes DISPLAYCOMPOSITE CLOSEDCURVE)
[roleset Order
(vr NUMBERP)]
[roleset Side
(mods Component@DISPLAYCOMPOSITE)
(number (3 NIL))
(vr LINESEGMEMT)]

(* Note that there is a possible redundancy in how the positions of the vertices are expressed.)

Source File: AIPSREALIZATION 114
There may be an Approximation Edgeset, or the positions may be roles of the Sides.)

(InitRECTANGLE
 [LAMBDA NIL

[concept RECTANGLE
   (specializes POLYGON)
   [roleset NIL
      (mods Order@POLYGON)
      (vr 4)]
   [roleset NIL
      (mods Side@POLYGON)
      (modality Optional)
      (number 4)]
   [roleset LowerLeft
      (diffs Location@POLYGON)
      (defaultfiller '(0 0))]
   [roleset UpperRight
      (diffs Location@POLYGON)
      (derivation (Copy (Prerequisites LowerLeft Height Width)
         (Binding (lowerLeft [$$CONCEPT;LowerLeft]:1))
         (Arguments <lowerLeft:1+([$$CONCEPT;Width]:1-1)
            lowerLeft:2+([$$CONCEPT;Height]:1-1) >))])

   [roleset NIL
      (mods Height@POLYGON)
      (derivation (Copy (Prerequisites LowerLeft UpperRight)
         (Arguments 1+[$$CONCEPT;UpperRight]:1:2
            -[$$CONCEPT;LowerLeft]:1:2])]

   [roleset NIL
      (mods Width@POLYGON)
      (derivation (Copy (Prerequisites LowerLeft UpperRight)
         (Arguments 1+[$$CONCEPT;UpperRight]:1:1
            -[$$CONCEPT;LowerLeft]:1:1])]

   (itags (ToDraw TDRectangle)
       (ToLocate TLRrectangle)
       (ToMake TMRrectangle)
       (ToSize TSRrectangle))])

(InitREGULARPOLYGON
 [LAMBDA NIL

(* Edited by J.Gibbons on 13-Jan-81.)
[concept REGULARPOLYGON
(specializes POLYGON)
[roleset Center
(mods Location@POLYGON)]
[roleset Radius
(mods Size@POLYGON)]
[roleset NIL
(mods Orientation@POLYGON)]
(itags (ToDraw TDRegularPolygon))]

{15}

(InitRealizationModel
[LAMBDA NIL
 (* Edited by J.Gibbons on
21-Jun-80.)

(* Initializes the AIPS Realization Model.
Concepts at this level have to do with geometric shapes
(which are isomorphic to descriptions of line drawings.))

(InitDisplayFormConcepts
 (* Eventually we will init the
bmg stuff here)

NIL])
)

(RPAQQ AIPSREALIZATIONTOLOCATEFNS (TLDisplayAtom TLRectangle))
(DEFINEQ

{16}

(TLDisplayAtom
[DLAMBDA ((displayAtom IndividualConcept (SATISFIES displayAtom df
`DISPLAYATOM))
(location (LST OF NUMBERP)))
 (* Edited by J.Gibbons on
29-Jan-81.)

(* TLDisplayAtom sets the Location Roleset of displayAtom to
location which defaults according to the Location's
DefaultFiller ITag.)

(if location
 then (SetRoleValues displayAtom Location@DISPLAYATOM <location>)
else location_{displayAtom;&Location}:l)
location)

{17}

Source File: AIPSREALIZATION 116
(TLRectangle
   [DLAMBDA ((rectangle IndividualConcept (SATISFIES rectangle df ~RECTANGLE))
                (location (LST OF SMALLP)))]
   (* Edited by J.Gibbons on 21-Dec-80.)
   (if location
        then (SetRoleValues rectangle LowerLeft@RECTANGLE <location>)
        else location_ {rectangle;&LowerLeft}:1)
   (SetRoleValues rectangle UpperRight@RECTANGLE
        <<location:1+({rectangle;&Width}:1-1)
        location:2+({rectangle;&Height}:1-1)
        >>)
   location))
)

(ADDTOVAR CKLONEFILES AIPSREALIZATION)

(RPAQQ AIPSREALIZATIONDERIVATIONFNS (MakeRectangle))
(DEFINEQ

{18}

(MakeRectangle
   [DLAMBDA ((height NUMBERP)
                (width NUMBERP))
                (* Make up a Rectangle from Height and Width information.)
                <(Make RECTANGLE Height height Width width)
                >]
)

(RPAQQ AIPSREALIZATIONTOMAKEFNS (TMRectangle))
(DEFINEQ

{19}

(TMRectangle
   [LAMBDA (genericRectangle)
                (* Edited by J.Gibbons on 18-Dec-80.)
                <(Make RECTANGLE)
                >]
)

(RPAQQ AIPSREALIZATIONTOSIZEFNS (TSRectangle))
(DEFINEQ

(TSRectangle
  [DLAMBDA ((rectangle IndividualConcept (SATISFIES rectangle df
    ~RECTANGLE))
    (height (ONEOF NIL SMALLP))
    (width (ONEOF NIL SMALLP)))
    (* Edited by J. Gibbons on 19-Dec-80.)
      (if height
        then (SetRoleValues rectangle Height@RECTANGLE <height>)
        else height_{rectangle;&Height}:1)
      (if width
        then (SetRoleValues rectangle Width@RECTANGLE <width>)
        else width_{rectangle;&Width}:1)
      <height width>])
)
STOP

Source File: AIPSREALIZATION  118
12. **SOURCE FILE: AIPSTABLE**

FindColumnTableRows ........................................... 10
FindColumnWidth ................................................. 11
FindRowHeight .................................................... 12
FindRowTableColumns ............................................ 13
FindTableEntityIndicators ..................................... 14
FindTableThemeElementIndicators .............................. 15
FindTableThemeElementIndicatorsAndOrientation .......... 16
InitARRANGEMENT ................................................. 1
InitCOLUMN ...................................................... 2
InitCOLUMNTABLE ................................................ 3
InitROW .......................................................... 4
InitROWTABLE .................................................... 5
InitTABLE ........................................................ 6
InitTableConcepts ................................................ 9
InitTEXTCOLUMN ................................................ 7
InitTEXTROW ...................................................... 8
MakeDefaultTableGround ...................................... 17
MakeTableColumnOrderList .................................... 18
MakeTableEntryColumns ...................................... 19
MakeTableEntryRows ......................................... 20
MakeTableFooter .............................................. 21
MakeTableHeaderRow ......................................... 22
MakeTableLeftIndex .......................................... 23
MakeTableRightIndex? ........................................ 24
MakeTableRowOrderList ...................................... 25
TLCColumnTable ................................................. 26
TLRowTable ...................................................... 27
changes to: FindColumnTableRows
previous date: "17-Dec-80 20:06:11" <NEWAIPS>AIPSTABLE..35)

(PRETTYCOMPRINT AIPSTABLECOMS)
(RPAQQ AIPSTABLECOMS ((FNS * AIPSTABLEINITFNS)
(FNS * AIPSTABLEDERIVATIONFNS)
(FNS * AIPSTABLETOLOCATEFNS)
(ADDVARS (CKNONEFILES AIPSTABLE))))

(RPAQQ AIPSTABLEINITFNS (InitARRANGEMENT InitCOLUMN InitCOLUMNTABLE InitROW
InitROWTABLE InitTABLE
InitTEXTCOLUMN InitTEXTROW
InitTableConcepts))

(DEFINEQ

[1]

(InitARRANGEMENT
[LAMBDA NIL (* Edited by F.Zdybel on 18-Jul-80.)
[concept ARRANGEMENT

(* An Arrangement is any collection of Display Items whether or not their arrangement conveys information. Rows, Columns, Legends are examples of Arrangements.)

(specializes DISPLAY)
[roleset Item
(diffs Realization@DISPLAY)
(modality Obligatory)
(number (2 NIL)])

[roleset ItemOrder
(vr LISTP)]

(* If there aren't at least two items, why bother?)

(* The list which fills this Role can be used to impose an ordering on the fillers of the Item Role.)
[roleset Border
 (diffs Realization@DISPLAY)
 (vr CLOSEDCURVE)]
[roleset Separator
 (diffs Realization@DISPLAY)
 (modality Optional)
 (number (0 NIL))
 (vr DISPLAYITEM)]

(* Could be anything but is probably a Line.))}

(InitCOLUMN
 [LAMBDA NIL (* Edited by F.Zdybel on 12-Dec-80.)
 [concept COLUMN

 (specializes ARRANGEMENT)
 [roleset NIL
 (mods Item@ARRANGEMENT)
 (modality Obligatory)
 (number (2 NIL))
 (vr DISPLAY)]
 [roleset NIL
 (mods Width@DISPLAYITEM)
 (derivation (FindColumnWidth (Prerequisites Item)
 (Arguments $$CONCEPT;Item)))))])

][3]

(InitCOLUMNTABLE
 [LAMBDA NIL (* Edited by F.Zdybel on 16-Dec-80.)
 [concept COLUMNTABLE

 (* A Table whose Theme is spread along its columns. Hence, a row of columns.)

 (specializes TABLE ROW)
 [roleset NIL
 (mods Ground@TABLE)
 (derivation (MakeDefaultTableGround))]
 [roleset NIL
 (mods ThemeList@TABLE)

 121 Source File: AIPSTABLE
(derivation (FindTableThemeElementIndicators (Prerequisites Application)
  (Arguments
  [$\$$CONCEPT; Application: ])))

(PROG [(column [roleset Column
  (mods Group@TABLE Item@ROW)
  (vr TEXTCOLUMN))]
  [roleset LeftIndex
   (diffs (atomval column))
   (modality Obligatory)
   (derivation (MakeTableLeftIndex (Arguments
     [$\$$CONCEPT; &ThemeList1 : 1
     (CAR
     [$\$$CONCEPT; &Ground]))
     (fillwhenmade))])

  (* The LeftIndex is the column of display labels that runs down the left edge of the ColumnTable.)

  [roleset RightIndex
   (diffs (atomval column))
   (derivation (MakeTableRightIndex? (Arguments
     (PLENGTH (CAR
     [$\$$CONCEPT; &EntityList]))
     (CAR
     [$\$$CONCEPT; &LeftIndex]))]
     (fillwhenmade))]

  (* A very wide Table might also want an index along its right edge.)

  [roleset Entry
   (diffs (atomval column))
   (modality Obligatory)
   (number (1 NIL))
   (derivation (MakeTableEntryColumns (Prerequisites Application)
     (Arguments
     (CAR
     [$\$$CONCEPT; &EntityList]))
     (CAR
     [$\$$CONCEPT; &ThemeList1])
     (CAR
     [$\$$CONCEPT; &Ground]))
     (fillwhenmade)]]

  [roleset NIL
   (mods ItemOrder@ARRANGEMENT)
   (derivation (MakeTableColumnOrderList (Arguments
     (CAR
     [$\$$CONCEPT; &LeftIndex])
     [$\$$CONCEPT; &Entry] (NLSETQ
     [$\$$CONCEPT; &RightIndex]: 1)))]

Source File: AIPSTABLE
(* While the derivation and realization of a ColumnTable are in terms of its constituent Columns, for layout purposes it is also necessary to describe the table in terms of rows. *)

(InitROW
 [LAMBDA NIL
  (* Edited by F.Zdybel on 12-Dec-80.)
  [concept ROW

  (* Note that a Row in a Table may have a non-linear structure (e.g., in order to "compress" the width of the table by putting two interleaved rows together to make a single group in the table.)

  (specializes ARRANGEMENT)
  [roleset NIL
   (mods Item@ARRANGEMENT)
   (modality Obligatory)
   (number (2 NIL))
   (vr DISPLAY)]
  [roleset NIL
   (mods Height@DISPLAYITEM)
   (derivation (FindRowHeight (Prerequisites Item)
    (Arguments {$$CONCEPT;Item$})))]])

[5]

(InitROWTABLE
 [LAMBDA NIL
  (* Edited by F.Zdybel on 12-Dec-80.)
  [Source File: AIPSTABLE]}]}
(specializes TABLE COLUMN)
roleset NIL
(mods Ground@TABLE)
(derivation (MakeDefaultTableGround))
roleset NIL
(mods ThemeList@TABLE)
(derivation (FindTableThemeElementIndicators (Prerequisites Application)
(Arguments

(PROG [\(\text{\texttt{row}}\ [\text{\texttt{roleset\ Row}}\)
(mods Group@TABLE Item@COLUMN)

([\text{\texttt{roleset\ Header}}\)

((\text{\texttt{diffs\ (atomval\ row)}})
(modality\ Obligatory)
(derivation\ (MakeTableHeaderRow\ (Arguments

[\text{\texttt{Header}}:1

(fillwhenmade)]

(*\ The\ Header\ is\ the\ row\ of\ display\ labels\ that\ runs\ across\ the\ top\ of\ the\ \text{\texttt{RowTable}}.)

[\text{\texttt{roleset\ Foot}}\)

((\text{\texttt{diffs\ (atomval\ row)}})
(derivation\ (MakeTableFoot?\ (Arguments\ (FLENGTH

[\text{\texttt{FLength}}:1

])

(*\ A\ very\ tall\ table\ might\ want\ another\ index\ running\ along\ the\ bottom\ edge.)

[\text{\texttt{roleset Entry}}\)

((\text{\texttt{diffs\ (atomval\ row)}})
(modality\ Obligatory)
(number\ (1\ NIL))
(derivation\ (MakeTableEntryRows\ (Arguments

[\text{\texttt{EntityList}}:1

Source File: AIPSTABLE 124
(fillwhenmade))
[roleset NIL
(mods ItemOrder@ARRANGEMENT)
(derivation (MakeTableRowOrderList (Arguments $$CONCEPT;&Header}:1
$$CONCEPT;&Entry}{
NLSETQ
$$CONCEPT;Foot}:1))
]
[roleset Column
(vr TEXTCOLUMN)
[derivation (FindRowTableColumns (Prerequisites Row)
(Arguments $$CONCEPT;&ItemOrder}:1
(FLENGTH $$CONCEPT;ThemeList}:1]

(* While the derivation and realization of a RowTable are in
terms of its constituent Rows, for layout purposes it is also
necessary to describe the table in terms of columns.)

(itags (ToLocate TLRowTable)
(NumEntriesJustifyingAFoot 10)
(HorizontalSeparation 10)
(VerticalSeparation 10)))

(InitTABLE
[LAMBDA NIL
[concept TABLE

(specializes ARRANGEMENT)
[roleset NIL
(mods Application@DISPLAY)
(number 1)]

(* For the moment we will restrict the Application Role of
table to take a single ITEmplate as filler, and the derivation
procedures will enforce this restriction via their DECL, even
if KL-ONE per se does not.)

[roleset Group

125 Source File: AIPSTABLE
(* Groups are either the columns or rows of the Table. Just which depends on how one wishes to describe the Table: as a set of columns or a set of rows.)

[role-set EntityList
  (vr LISTP)
  (derivation (FindTableEntityIndicators (Prerequisites Application)
    (Arguments {$CONCEPT;Application}))))

[role-set ThemeList
  (vr LISTP)
  (derivation (FindTableThemeElementIndicatorsAndOrientation
    (Prerequisites Application)
    (Arguments {$CONCEPT;Application}
      {$CONCEPT;&EntityList:1})))
  (fill-when-made)]

(* A list of the highest level slots (domain world Roles) indicated in the ITemplates filling the application Role. The ordering corresponds to the ordering of the entries in each group of the eventual Table.)

])]

(InitTEXTCOLUMN
  [LAMBDA NIL]
  (* Edited by F.Zdybel on 25-Jul-80.)
  [concept TEXTCOLUMN
    (* A Column whose items are Texts.)
    (specializes COLUMN)
    [role-set NIL
      (mods Item@COLUMN)
      (vr TEXT)])]

(InitTEXTROW
  [LAMBDA NIL]
  (* Edited by F.Zdybel on 25-Jul-80.)
  [concept TEXTROW
    Source File: AIPSTABLE 126]
(specializes ROW)
[roleset NIL
(mods Item@ROW)
(vr TEXT)]]]}}

{9}

(InitTableConcepts
[LAMBDA NIL

(InitARRANGEMENT)
(InitROW)
(InitCOLUMN)
(InitTEXTROW)
(InitTEXTCOLUMN)
(InitTABLE)
(InitROWTABLE)
(InitCOLUMNTABLE))

(RPAQQ AIPSTABLEDERIVATIONFNS (FindColumnTableRows FindColumnWidth
FindRowHeight
FindRowTableColumns
FindTableEntityIndicators
FindTableThemeElementIndicators
FindTableThemeElementIndicatorsAndOrientation
MakeDefaultTableGround
MakeTableColumnOrderList
MakeTableEntryColumns
MakeTableEntryRows
MakeTableFoot?
MakeTableHeaderRow
MakeTableLeftIndex
MakeTableRightIndex?
MakeTableRowOrderList))

(DEFINEQ

{10}

(FindColumnTableRows
[DLAMBDA ((columnList (LST OF IndividualConcept
(SATISFIES (HomogenousConceptListP columnList
`COLUMN))))

(numRows FIXP)
(RETURNS (LST OF IndividualConcept (SATISFIES

Source File: AIPSTABLE
(HomogenousConceptListP VALUE:1 'ROW)))})
(* Edited by F.Zdybel on 18-Dec-80.)

(* Figures out the rows of a table that has been built in terms of columns.)

(for rowNum from 1 to numRows bind itemOrderList
 collect (Make ROW Item itemOrderList_(for columnDescr in columnList
 collect (FNTH
 {columnDescr; ItemOrder1:1
 rowNum):1
 ItemOrder itemOrderList)))
)

(FindColumnWidth
[DLAMBDA ((itemList (LST OF IndividualConcept
 (SATISFIES (HomogenousConceptListP itemList
 'DISPLAY))))
 (RETURNS (LST OF FIXP)))
 (* Edited by F.Zdybel on 12-Dec-80.)

(* Determines the width of a column of Displays by finding the maximum width of the items.)

<(APPLY 'MAX {itemList;&Width})
 >])

(FindRowHeight
[DLAMBDA ((itemList (LST OF IndividualConcept
 (SATISFIES (HomogenousConceptListP itemList
 'DISPLAY))))
 (RETURNS (LST OF FIXP)))
 (* Edited by F.Zdybel on 12-Dec-80.)

(* Determines the height of a row of Displays by finding the maximum height of the items.)

<(APPLY 'MAX {itemList;&Height})
 >])

Source File: AIPSTABLE 128
(FindRowTableColumns
 DLAMBDAD ((rowList (LST OF IndividualConcept (SATISFIES (HomogenousConceptListP rowList 'ROW')))
 (numColumns FIXP)
 (RETURNS (LST OF IndividualConcept
 (SATISFIES (HomogenousConceptListP VALUE:1 'COLUMN')))))

(* Edited by F.Zdybel on 12-Dec-80.)

(* Figures out the columns of a table that has been built in terms of rows.)

(for columnNum from 1 to numColumns bind itemOrderList
 collect (Make COLUMN Item itemOrderList_ (for rowDescr in rowList
 collect (FNTH {rowDescr;ItemOrder}:1 columnNum):1)
 ItemOrder itemOrderList))))

(FindTableEntityIndicators
 DLAMBDAD ((tableApplicationList (LST OF IndividualConcept) (SATISFIES
 tableApplicationList:1 df 'ITEMPLATE and ~tableApplicationList:1))
 (RETURNS (LST OF (LST OF IndividualConcept)) (SATISFIES
 HomogenousConceptListP VALUE:1 'CONCEPT'))))

(* Edited by F.Zdybel on 4-Dec-80.)

(* Gets the ConceptGroup fillers of the template filling the Application Role.)

< {tableApplicationList:1;ConceptGroup} >])

(FindTableThemeElementIndicators
 DLAMBDAD ((tableApplicationList (LST OF IndividualConcept)
 (SATISFIES tableApplicationList:1 df 'ITEMPLATE
 and ~tableApplicationList:1))
 (RETURNS (LST OF (LST OF IndividualConcept)
 (SATISFIES HomogenousConceptListP VALUE:1 'ROLE'))))
(* Gathers together the roles meta-indicated by the ITemplate filling the Application Role. Gives special treatment to any Role name Name, by placing it first in the list.)

(PROG ((themeRoleIndicatorList [tableApplicationList:1;RoleGroup])
  nameRoleIndicator)
  (if nameRoleIndicator-
    in themeRoleIndicatorList
    thereis
      ('Name)
      fmemb
      (KLFindNamesOfRole
        (KLGetMetaDescribedEntity roleIndicator
          (KLGetNamedContext 'AIPSDOMAINMETA)
      )
    )
    then themeRoleIndicatorList
  <nameRoleIndicator ! (DREMOVE nameRoleIndicator thèmeRoleIndicatorList)
  >)
  (RETURN < themeRoleIndicatorList >)))

(FindTableThemeElementIndicatorsAndOrientation
 [DLAMBDA ((tableApplicationList (LST OF IndividualConcept)
            (SATISFIES (tableApplicationList:1 df 'ITEMPLE
             and ~(tableApplicationList::1)))
            (tableEntityIndicatorList (LST OF IndividualConcept)
              (SATISFIES HomogenousConceptListP
                tableEntityIndicatorList 'CONCEPT)))
          (RETURNS [LST OF (LST OF IndividualConcept
            (SATISFIES (HomogenousConceptListP
              VALUE:1
              'ROLE)))
        (* Edited by F.Zdybel on 4-Dec-80.

(* Gathers together the role meta-indicators on the ITemplate filling the Application Role. Gives special treatment to the indicator of any Role named Name, by placing it first in the list. Finally compares the number of entities in the table with the number of theme elements and decides whether the Table should become a RowTable or a ColumnTable.)
(PROG ((themeRoleIndicatorList {tableApplicationList:1;RoleGroup}))
  nameRoleIndicator temp)
  (if nameRoleIndicator_ (for roleIndicator
    in themeRoleIndicatorList
    thereis
      ('Name)
      fmemb
      (KFindNamesOfRole
        (KLGetMetaDescribedEntity
          roleIndicator
          (KLGetNamedContext
            'AIPSDOMAINMETA)
      then themeRoleIndicatorList_ _)
    nameRoleIndicator !_ (DREMOVE nameRoleIndicator
      themeRoleIndicatorList))
  (if (FLENGTH tableEntityIndicatorList) lt temp_ (FLENGTH
    themeRoleIndicatorList)
    and temp gt 3
    then $$PROTOTYPE 'COLUMNTABLE
      else $$PROTOTYPE 'ROWTABLE)
  (RETURN < themeRoleIndicatorList >)))

{17}

(MakeDefaultTableGround
  [DLAMBDA ((RETURNS (LST OF IndividualConcept) (SATISFIES (VALUE:1 df
    'VIEWSURFACE))))
    (* Edited by F.Zdybel on 13-Nov-80.)

(* Make a ViewSurface of random size so that at least all of
the Displays in the Table can have their Ground roles filled.
The dimensions of the ViewSurface can be changed later.)

< (Make VIEWSURFACE Continuum (Make CARTESIANSYSTEM)
  Boundary
  (Make RECTANGLE LowerLeft <0 0> UpperRight <100 100>)
>

{18}

(MakeTableColumnOrderList
  [DLAMBDA ((leftIndex IndividualConcept
    (SATISFIES (leftIndex df 'COLUMN)))
    (entryColumnList (LST OF IndividualConcept)
      (SATISFIES HomogenousConceptListP entryColumnList
        'COLUMN)))
    (rightIndex (ONEOF (IndividualConcept

131 Source File: AIPSTABLE
(SATISFIES (rightIndex df 'COLUMN)))
NIL)
(RETURNS (LST OF (LST OF IndividualConcept))
(SATISFIES HomogenousConceptListP VALUE:1 'COLUMN))))
(* Edited by F.Zdybel on 13-Nov-80.)

(* Produces the list that imposes the correct ordering on the
Columns of the ColumnTable. Hopefully entryColumnList already
reflects the ordering embedded in the ColumnTable's
Application.)

(PROG ((result (<leftIndex ! entryColumnList>>)>)
(if rightIndex
then result_ < ! ! result rightIndex>)
(RETURN <result>)))

(MakeTableEntryColumns
(DLAMBDA ((tableEntityList (LST OF IndividualConcept)
(SATISFIES (HomogenousConceptListP tableEntityList
'CONCEPT)))))
(tableThemeList (LST OF IndividualConcept)
(SATISFIES (HomogenousConceptListP tableThemeList
'ROLE))))
(tableGround IndividualConcept (SATISFIES tableGround df
'VIEWSURFACE))
(RETURNS (LST OF IndividualConcept) (SATISFIES
(HomogenousConceptListP VALUE 'COLUMN))))
(* Edited by F.Zdybel on 4-Dec-80.)

(* Produces the Columns in a ColumnTable;
one Column for each element of EntityList.
Each Column contains one Item for each element in ThemeList.
Fills the ItemOrder Role of the Table to indicate the ordering
of the Columns, and fills the ItemOrder Role of each Column to
indicate the ordering of the items in the Column.)

(for entityIndicator in tableEntityList
bind itemOrder (itemPrototype _ (KLGetConceptName (KLFindVRsOfRole
Item@{$$PROTOTYPE;Entry}:1):1))
collect (Make TEXTCOLUMN Application
(Make ITEMPART ConceptGroup entityIndicator RoleGroup
tableThemeList))

Source File: AIPSTABLE 132
Item itemOrder_(for themeElementIndicator
in tableThemeList
collect (APPLY* 'Make itemPrototype
'Application
(Make ITEMPLATE
ConceptGroup
entityIndicator
RoleGroup
themeElementIndicator)
'Ground tableGround))
ItemOrder itemOrder Ground tableGround))})

{20}

(MakeTableEntryRows
[DLAMBDA ((tableEntityList (LST OF IndividualConcept)
(SATISFIES (HomogenousConceptListP tableEntityList
'CONCEPT)))
(tableThemeList (LST OF IndividualConcept)
(SATISFIES (HomogenousConceptListP tableThemeList
'ROLE)))
(tableGround IndividualConcept (SATISFIES tableGround df
'VIEWSURFACE))
(RETURNS (LST OF IndividualConcept) (SATISFIES
(HomogenousConceptListP
VALUE 'ROW))))
(* Edited by F.Zdybel on
4-Dec-80.)

(* Produces the Rows in a RowTable; one Row for each element
of EntityList. Each Row contains one Item for each element in
ThemeList. Fills the ItemOrder Role of the Table to indicate
the ordering of the Rows, and fills the ItemOrder Role of each
Row to indicate the ordering of the items in the Row.)

(for entityIndicator in tableEntityList
bind itemOrder (itemPrototype _ (KLGetConceptName (KLFindVRsOfRole
Item@{$SPROTOTYPE;Entry}:1):1))
collect (Make TEXTROW Application (Make ITEMPLATE ConceptGroup
entityIndicator RoleGroup
tableThemeList)
Item itemOrder_(for themeElementIndicator
in tableThemeList
collect (APPLY* 'Make itemPrototype
'Application
(Make ITEMPLATE
ConceptGroup
entityIndicator

133 Source File: AIPSTABLE
RoleGroup
themeElementIndicator)
'Ground tableGround))

ItemOrder itemOrder Ground tableGround))

{21}

(MakeTableFoot?
[DLAMBDA ((numberOfRows NUMBERP)
  (tableHeader IndividualConcept
   (SATISFIES tableHeader df 'ROW))
   (RETURNS (LST OF IndividualConcept)
     (SATISFIES VALUE:1 df 'ROW)))
   (* Edited by F.Zdybel on 13-Nov-80.)

(* Decides whether or not to create a Foot for a RowTable by
  comparing the numberOfRows with the value of the
  NumEntriesJustifyingAFoot tag. The Foot, if created, is a copy
  of the Header.)

(PROG ((tableGround (CAR {tablefleader;Ground})))
  (if [IGEQ numberOfRows (CAR (KLFindIaData P$PCONCEPT
  NumEntriesJustifyingAFoot]
    then (RETURN
      <[Make TEXTROW Ground tableGround Item
        {for headerItem in {tableHeader;Item}
          collect (Make TEXT Ground tableGround String
            (CAR {headerItem;String})
          )
        ]
      >)
    else (StopFill))))))

{22}

(MakeTableHeaderRow
[DLAMBDA ((tableThemeList (LST OF IndividualConcept)
  (SATISFIES (HomogenousConceptListP tableThemeList
    'ROLE))
  (tableGround IndividualConcept (SATISFIES tableGround df
    'VIEWSURFACE))
  (RETURNS (LST OF IndividualConcept)
    (SATISFIES VALUE:1 df 'ROW))
  (* Edited by F.Zdybel on 7-Dec-80.)

(* Derives the Header Row for a RowTable, by going through the
  Roles meta-indicated by the elements of themeList and taking
  the most specific names of those roles.)

Source File: AIPSTABLE 134
(MakeTableLeftIndex
[DLAMBDA ((tableThemeList (LST OF IndividualConcept)
(SATISFIES (HomogenousConceptListP tableThemeList
ROLE)))
(tableGround IndividualConcept (SATISFIES tableGround df
"VIEWSURFACE"))
(RETURNS (LST OF IndividualConcept) (SATISFIES VALUE;1 df
"COLUMN")))
(* Edited by F.Zdybel on 7-Dec-80.)

(* Derives the LeftIndex for a ColumnTable, by going through
the Roles meta-indicated by the elements of themeList and
taking the most specific names of those roles.)

<(Make TEXTCOLUMN Item itemOrder_ (for roleMetaIndicator
in tableThemeList
collect
(Make
TEXT String
[MKSTRING
(CAR
(KLPfindNamesOfRole
(KLGetMetaDescribedEntity
roleMetaIndicator
(KLGetNamedContext
'AIPSDOMAINMETA
Ground tableGround))
ItemOrder itemOrder Ground tableGround))
>])

135 Source File: AIPSTABLE
(MakeTableRightIndex?
[DLAMBDA ((numberOfColumns NUMBERP)
   (tableLeftIndex IndividualConcept (SATISFIES tableLeftIndex df 'COLUMN))
   (RETURNS (LST OF IndividualConcept) (SATISFIES VALUE:1 df 'COLUMN))
   (* Edited by F.Zdybel on 13-Nov-80.)

(* Decides whether or not to create a RightIndex for a ColumnTable by comparing the numberOfColumns with the value of the NumEntriesJustifyingARightIndex tag. The RightIndex, if created, is a copy of the LeftIndex.)

(PROG ((tableGround (CAR {tableLeftIndex;Ground})))
   (if [IGEQ numberOfColumns (CAR (KLFindIData $$CONCEPT NumEntriesJustifyingARightIndex)
       then (RETURN
       <![Make TEXTCOLUMN Ground tableGround Item
         for leftIndexItem in {tableLeftIndex;Item}
         collect (Make TEXT Ground tableGround String
            (CAR {leftIndexItem;String}))
       ])
   else (StopFill))))

(MakeTableRowOrderList
[DLAMBDA ((headerRow IndividualConcept (SATISFIES (headerRow df 'ROW)))
   (entryRowList (LST OF IndividualConcept)
      (SATISFIES (HomogenousConceptListP entryRowList 'ROW))
   (footRow (ONEOF (IndividualConcept
      (SATISFIES (footRow df 'ROW)))
      NIL))
   (RETURNS (LST OF (LST OF IndividualConcept))
      (SATISFIES (HomogenousConceptListP VALUE:1 'ROW))))
   (* Edited by F.Zdybel on 13-Nov-80.)

(* Produces the list that imposes the correct ordering on the Rows of the RowTable. Hopefully entryRowList already reflects the ordering embedded in the RowTable's Application.)

Source File: AIPSTABLE
(PROG ((result (<headerRow ! < ! entryRowList>)))
  (if footRow
      then result_ < !! result footRow>)
  (RETURN <result>))))
)

(RPAQQ AIPSTABLETOLOCATEFNS (TLColumnTable TLRowTable))
(DEFINEQ

(TLColumnTable
 [DLAMBDA ((tableDescr IndividualConcept (SATISFIES tableDescr df
 `COLUMNTABLE))
   (tableLoc (ONEOF NIL (LST OF FIXP (SATISFIES "tableLoc::2")))))))

(* Edited by F.Zdybel on 17-Dec-80. *)

(* Locates the interior elements of a ColumnTable, given the
direction of the ColumnTable’s lower left corner.
If no location is given, the location
(0,0) is assumed)

(if ~tableLoc
  then tableLoc ’(0 0))
(for columnDescr in (CAR [tableDescr;&ItemOrder])
  bind (currentX _ tableLoc:1)
  currentY
  (bottomY _ tableLoc:2)
  [vertSeparation _ (CAR (KFLFindIData tableDescr 'VerticalSeparation)]
  [horizSeparation _ (CAR (KFLFindIData tableDescr 'HorizontalSeparation)]
  (rowHeightList _ [tableDescr;&Row;&Height])
  do (currentY currentY+rowHeight+vertSeparation)
  (SendMessage columnDescr 'ToLocate <currentX currentY>)
  (for itemDescr in (REVERSE (CAR [columnDescr;&ItemOrder]))
    as rowHeight in (REVERSE rowHeightList)
    do (SendMessage itemDescr 'ToLocate <currentX currentY>)
    (currentX currentX+(CAR [columnDescr;&Width])+horizSeparation)
  finally (KLC~iangeRoleValue (KL~indOneNamedlnstanceRole (CAR
    [tableDescr;Ground;Boundary]))
     'UpperRight)
  <currentX currentY>)))

[26]  [27]
(TLRowTable
[DLAMBDA ((tableDescr IndividualConcept (SATISFIES (tableDescr df `ROWTABLE)))
  (tableLoc (ONEOF NIL (LST OF FIXP (SATISFIES `tableLoc::2))))))
(* Edited by F.Zdybel on 17-Dec-80.)

(* Locates the interior elements of a RowTable, given the location of the RowTable's lower left corner. If no location is given, the location (0,0) is assumed.)

(if `tableLoc then tableLoc '(0 0))
(for rowDescr in (REVERSE {tableDescr;&ItemOrder}:l)
  bind (leftX tableLoc:l)
  currentX
  (currentY tableLoc:2)
  [vertSeparition _ (CAR (KLFindIData tableDescr '(VerticalSeparation))]
  [horizSeparition _ (CAR (KLFindIData tableDescr '(HorizontalSeparation))]
  (columnWidthList _ {tableDescr;&Column;&Width})
  do currentX leftX
  (SendMessage rowDescr 'ToLocate <currentX currentY>)
  (for itemDescr in {rowDescr;&ItemOrder}:l as columnWidth
   in columnWidthList do (SendMessage itemDescr 'ToLocate
    <currentX currentY>)
    currentX currentX+columnWidth+horizSeparation)
  currentY currentY+{rowDescr;&Height}:l+vertSeparition
  finally (KLChangeRoleValue (KLFindOneNamedInstanceRole (CAR
    {tableDescr;Ground;Boundary})
    'UpperRight)
    <currentX currentY>))])

(ADDTOVAR CklOnefiles AIPSTABLE)
STOP

Source File: AIPSTABLE 138
13. SOURCE FILE: AIPSTEMP

FindMostSpecializedConcept......17
HomogenousConceptListP.........18
InitAIPS........................1
InitDISPLAYITEM.................2
InitDISPLAYITEMABSTRACTION....3
InitDISPLAYSURFACE..............4
InitHOMOGENEOUSREGION..........5
InitINVISIBLESURFACE............6
InitMiddleEarth................8
InitMOTILEDISPLAYITEM.........7
InitPLANE SURFACE..............9
InitRECTANGULARREGION.........10
InitSTABILEDISPLAYITEM........11
InitUSERMODEL..................12
InitUserModel..................13
InitViewingOrganizationModel...16
InitVIEWSURFACE.................14
InitVISIBLESURFACE............15
TLDisplayItem..................19
TLDisplaySurface..............20
TLPlaneSurface.................21
TMDisplaySurface..............22
TMRegionNumber................23
TSPlaneSurface................24
TSRectangularRegion...........25
changes to: InitDISPLAYSURFACE APISTEMPCOMS AIPSTEMPUTILITYFNS
AIPSTEMPTOMAKEFNS TRegionNumber

previous date: "31-Jan-81 04:23:25" <NEWAIPS>AIPSTEMP..110)

(PRETTYCOMPRINT APISTEMPCOMS)

(RPAQQ APISTEMPCOMS [(FNS * APISTEMPINITFNS)
(FNS * APISTEMPUTILITYFNS)
(FNS * APISTEMPTOLOCATEFNS)
(FNS * APISTEMPTOMAKEFNS)
(FNS * APISTEMPTOSIZEFNS)
(DECLARE: DONTEVAL@LOAD EVAL@COMPILEWHEN
(NOT (BOUNDP (QUOTE BMGCOMS))))
DONTCOPY
(FILES <FONTWORK>BMGRECORDSANDVARS..0))
(ADDVARS (CKLONEFILES AIPSTEMP))
(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
COMPILEVARS (ADDVARS (NLAMA)
(NLAML)
(LAMA))

(RPAQQ APISTEMPINITFNS (InitAIPS InitDISPLAYITEM InitDISPLAYITEMABSTRACTION
InitDISPLAYSURFACE InitHOMOGENEOUSREGION
InitINVISIBLESURFACE InitMOTILEDISPLAYITEM
InitMiddleEarth InitPLANESURFACE
InitRECTANGULARREGION
InitSTABILEDISPLAYITEM InitUSERMODEL
InitUserModel InitVIEWSURFACE
InitVISIBLESURFACE
InitViewingOrganizationModel))

(DEFINEQ

(InitAIPS
[ LAMBDA NIL

(* Edited by J.Gibbons on 11-Nov-80.)
(* Initializes the Presentation, Viewing Organization and Realization Models of AIPS.)

(InitDISPLAYITEM)
(InitDISPLAYITEMABSTRACTION)
(InitUserModel)

Source File: AIPSTEMP
(InitCoordinateSystemConcepts)
(InitTimeSystemConcepts)
(InitCommonDomainModel)
(InitNavalDomainModel)
(InitRealizationModel)
(InitPresentationModel)
(InitNTDSConcepts)
(InitViewingOrganizationModel)
(InitDeviceConcepts)

(InitDISPLAYITEM
[LAMBDA NIL
(concept DISPLAYITEM

(* A Display Item can be either a Presentation or a Display Form. Thus decomposition of Presentations occurs until the level of Display Forms is reached.)

[roleset Location
 (vr LISTP)
 (defaultfiller <0 0>)]

(* Relates the location of the origin of the abstract figure (in the case of Displays and DisplayComposites) or is further differentiated or modified in the case of DisplayAtoms (e.g.: the EndPoint Role of LINE.))

[roleset Size
 (vr NUMBERP)
 (defaultfiller 1)]

(* Indicates a scaling transformation which is applied to the abstract figure before translation and after rotation (in the case of Displays and DisplayComposites) or is further modified or differentiated in the case of DisplayAtoms (e.g.: the SemiMajorAxis and SemiMinorAxis Roles of ELLIPSE.) The filler is always a positive number greater than or equal to one since we do not ever wish to scale abstract figures down, and in the case of DisplayAtoms we are specifying size information in terms of pixels.)
[roleset Orientation
  (vr NUMBERP)
  (defaultfiller 0)]

(* Indicates a rotational transformation which is applied to
the abstract figure before translation and scaling
(in the case of Displays and DisplayComposites) or is further
modified in the case of DisplayAtoms (e.g.: the Orientation
Role of ELLIPSE.) The filler is expected to indicate in terms
of a number of degrees between -180 and 180 inclusive.)

[roleset Width
  (vr NUMBERP)]
[roleset Height
  (vr NUMBERP)]

(* These Roles are used by layout processes that operate at
the level of the Viewing Organization Model.)

[roleset Envelope
  (vr CLOSEDCURVE)]

(* This role is useful for characterizing the area subsumed by
a Display Item.)

(itags (ToLocate TLDisplayItem)))

(InitDISPLAYITEMABSTRACTION
  [LAMBDA NIL
    [concept DISPLAYITEMABSTRACTION
      (specializes ABSTRACTION)
      [roleset FixedComponent
        (number (1 NIL))
        (vr DISPLAYFORM)]

(* Edited by J.Gibbons on
11-Nov-80.)

(* Abstractions of DisplayItems
are individuals of this
concept.)
(* Any visible components of the DisplayItem that are fixed in terms of identity and location (e.g.: the circle and dot of an NTDS friendly ship symbol.))

(itags (ToDraw TDDisplayItemAbstraction)))

(InitDISPLAYSURFACE
 [LAMBDA NIL (* Edited by J.Gibbons on 2-Feb-81.)

[concept DISPLAYSURFACE

(* A Viewsurface that represents an actual chunk of bitmap memory, either visible or invisible.)

(specializes DISPLAYATOM VIEWSURFACE)
[roleset Plane
  (modality Obligatory)
  (vr PLANESURFACE)]

(* Indicates which of the primal Display Surfaces the Display Surface is located on.)

[roleset NIL
  (mods Height@DISPLAYATOM)
  (mods Height@VIEWSURFACE)]
[roleset NIL
  (mods Width@DISPLAYATOM)
  (mods Width@VIEWSURFACE)]
[roleset RegionNumber
  (vr NUMBERP)
  (derivation (TMRegionNumber (Prerequisites Plane Location Height
                                Width Background)
                                (Arguments [$$CONCEPT;Plane]:1
                                $$CONCEPT;Location]:1
                                $$CONCEPT;Height]:1
                                $$CONCEPT;Width]:1
                                $$CONCEPT;Background]:1))))

(* Establishes the correspondence with a BMG Display Region more directly than via Nexii.)
[roleset Background
(modality Obligatory)
(vr SMALLP)
(defaultfiller 0)]

(* Sets the Background shade of the DisplaySurface.)

[roleset NIL
(mods Continuum@VIEWSURFACE)
(vr VIEWSYSTEM)]

(* Because a DISPLAYSURFACE represents actual map memory, the coordinate system must be integer Cartesian.)

(itags (ToDraw TDDisplaySurface)
(ToLocate TLDisplaySurface)
(ToMake TMDisplaySurface))]

(InitHOMOGENEOUSREGION
[LAMBDA NIL
[concept HOMOGENEOUSREGION
(specializes REGION)
[roleset NIL
(mods Aperture)
(number 0)]]])

(InitINVISIBLESURFACE
[LAMBDA NIL
[concept INVISIBLESURFACE
(* Edited by J.Gibbons on 5-Sep-80.)
(specializes DISPLAYSURFACE)
[roleset NIL
(mods Plane@DISPLAYSURFACE)
(vr [iconcept HIDDENPLANE of PLANESURFACE]]))

Source File: AIPSTEMP 144
(InitMOTILEDISPLAYITEM
 [LAMBDA NIL
 (* Edited by J.Gibbons on 5-Sep-80.)
 [concept MOTILEDISPLAYITEM

 (* The purpose of this concept is to mark which Display Items can be re-located (those whose locations are not significant to their semantics.))

 (specializes DISPLAYITEM))])

(InitMiddleEarth
 [LAMBDA NIL
 (* Edited by F.Zdybel on 31-Jan-80.)

 (* Initializes AIPS and the necessary Domain World Model SI-Net fragments for the Middle Earth demonstration system.)

 (InitAIPS)
 (InitGeographyModel))

(InitPLANE SURFACE
 [LAMBDA NIL
 (* Edited by J.Gibbons on 31-Jan-81.)

 [concept PLANE SURFACE
 (* Represents a plane of BMG-11 bitmap memory.)

 (specializes DISPLAYSURFACE)
 [role set PlaneNumber
 (modality Obligatory)
 (vr NUMBERP)]
 [role set NIL
 (mods Location@DISPLAYSURFACE)
 (derivation (Copy (Prerequisites PlaneNumber))]

Source File: AIPSTEMP
(Binding (specListRecord (BMGShowPlane
  [$$CONCEPT;PlaneNumber]:1)))
  (Arguments <specListRecord:6 specListRecord:10>))
]

[roleset NIL ]
(mods Height@DISPLAYSURFACE)
(derivation (Copy (Prerequisites PlaneNumber)
  (Binding (specListRecord (BMGShowPlane
    [$$CONCEPT;PlaneNumber]:1)))
  (Arguments 1+specListRecord:12-specListRecord:10))
)

[roleset NIL ]
(mods Width@DISPLAYSURFACE)
(derivation (Copy (Prerequisites PlaneNumber)
  (Binding (specListRecord (BMGShowPlane
    [$$CONCEPT;PlaneNumber]:1)))
  (Arguments 1+specListRecord:8-specListRecord:6)))

(itags (ToLocate TLPlaneSurface)
  (ToSize TSPlaneSurface)))

(InitRECTANGULARREGION
 [LAMBDA NIL
  (* Edited by J.Gibbons on
  19-Dec-80.)

[concept RECTANGULARREGION
  (specializes HOMOGENEOUSREGION)
[roleset NIL
  (mods Boundary@HOMOGENEOUSREGION)
  (modality Obligatory)
  (vr RECTANGLE)
  (derivation (TMRectangle))]
[roleset NIL
  (mods Continuum@HOMOGENEOUSREGION)
  (vr CARTESIANSYSTEM)]
[roleset NIL
  (mods Height@HOMOGENEOUSREGION)
  (derivation (Copy (Prerequisites Boundary)
    (Arguments [$$CONCEPT;Boundary;Height]:1)))]
[roleset NIL
  (mods Width@HOMOGENEOUSREGION)
  (derivation (Copy (Prerequisites Boundary)
    (Arguments [$$CONCEPT;Boundary;Width]:1)))]

(itags (ToSize TSRectangularRegion)))]

Source File: AIPSTEMP
(InitSTABILEDISPLAYITEM
 [LAMBDA NIL
   (* Edited by J.Gibbons on 5-Sep-80.)
   [concept STABILEDISPLAYITEM
    (* The purpose of this concept is to mark which display items cannot be re-located (those whose locations are significant to their semantics.))
    (specializes DISPLAYITEM)])]

(InitUSERMODEL
 [LAMBDA NIL
   (* Edited by J.Gibbons on 13-Nov-80.)
   (* A simple collection of parameters reflecting some of a user's preferences regarding system behavior.)
   [concept USERMODEL
    [roleset LabelLocality
     (fillwhenmade)
     (vr ATOM)
     (defaultfiller 'TOP)]
    [roleset ApertureHeight
     (fillwhenmade)
     (vr SMALLP)
     (defaultfiller 256)]
    [roleset ApertureWidth
     (fillwhenmade)
     (vr SMALLP)
     (defaultfiller 256)])]

(InitUserModel
 [LAMBDA NIL
   (* Edited by J.Gibbons on 24-Sep-80.)
   (* Defines the UserModel concept and instantiates a default model.)
   (InitUSERMODEL)
   (AIPSUSERMODEL_(Make USERMODEL)])]
(InitVIEWSURFACE
 [LAMBDA NIL

 [concept VIEWSURFACE

 (* A View Surface is a medium for the arrangement of viewable objects.)

 (specializes RECTANGULARREGION)
 [roleset Window
   (number (0 NIL))
   (vr WINDOW)]

 (* An aspect of View Surfaces is that Windows open onto them.)]])

(InitVISIBLESURFACE
 [LAMBDA NIL

 [concept VISIBLESURFACE

 (* Here the medium is constrained to be a Display Region onto one of the visible planes. This difference is reflected in a different derivation of the Display Region filling the Medium Role.)

 (specializes DISPLAYSURFACE)
 [roleset NIL
   (mods Plane@DISPLAYSURFACE)
   (itags (DefaultFiller "CONRACSCREEN"))]]]

(InitViewingOrganizationModel
 [LAMBDA NIL

 (* Initializes the Viewing Organization Model of the AIPS SI-Net. These concepts have to do with the syntactic

 Source File: AIPSTEMP

 148)
organization of display elements, and include such as Window, View Surface, Display Form, etc.)

(InitMOTILEDISPLAYITEM)
(InitSTABLEDISPLAYITEM)

(* Set up all the necessary concepts for describing positions on View Surfaces, Display Surfaces, etc.)

(InitHOMOGENEOUSREGION)
(InitRECTANGULARREGION)
(InitVIEWSURFACE)
(InitDISPLAYSURFACE)
(InitVISIBLESURFACE)
(InitINVISIBLESURFACE)
(InitPLANESURFACE) (* Sets up the generic descriptions of the different types of Windows.)

(InitWindowConcepts)
)

(RPAQQ AIPSTEMPUTILITYFNS (FindMostSpecializedConcept HomogenousConceptListP))

(*) Edited by F.Zdybel on 3-Jul-80.

(* Tries to find the concept that is at the lowest level of the hierarchy. Picks the first concept on conceptList that has no ancestors on conceptList.)

(FindMostSpecializedConcept
[DLAMBDA ((conceptList (LST OF GenericConcept))
 (RETURNS GenericConcept))

 (* Edited by F.Zdybel on 3-Jul-80.)

(*) Tries to find the concept that is at the lowest level of the hierarchy. Picks the first concept on conceptList that has no ancestors on conceptList.)

(for concept in conceptList thereis (for otherConcept in conceptList
 when (concept"=otherConcept)
 never (otherConcept df concept)))

Source File: AIPSTEMP
(HomogenousConceptListP
  [LAMBDA (cList restrictionC)
   (for element in cList always element df restrictionC)])

(RPAQQ AIPSTEMPTOLOCATEFNs (TLDisplayItem TLDisplaySurface TLPlaneSurface))

(DEFINEQ

(TLDisplayItem
  [DLAMBDA ((displayItemDescr IndividualConcept (SATISFIES
    (displayItemDescr df
     'DISPLAYITEM))
     (itemLoc (LISTP OF FIXP (SATISFIES "itemLoc::2")))))
   (* Edited by F.2dybel on
    16-Dec-80.)
   (* Simply locates the
    DisplayItem at the specified
    location.)
   (PROG (currentLocationRole)
    (if currentLocationRole (_KLFIndOneNamedlnstanceRole
      displayItemDescr
      'Location)
     then (_KLCChanceRoleValue currentLocationRole itemLoc)
     else (_KLSatisfyRole Location@DISPLAYITEM displayItemDescr
      <itemLoc>)))

(TLDisplaySurface
  [DLAMBDA ((displaySurface IndividualConcept (SATISFIES displaySurface df
    'DISPLAYSURFACE))
    (location (LST OF FIXP)))
   (* Edited by J.Gibbons on
    31-Jan-81.)
   (if location or "location_[displaySurface;Location]:1
    then (if location
     then (SetRoleValues displaySurface Location@DISPLAYSURFACE
      <location>)
     (PROG ((regionNumber [displaySurface;RegionNumber]:1))
      (if regionNumber
       then (BMGFreeRegionNumber regionNumber)
       (SetRoleValues displaySurface
        RegionNumber@DISPLAYSURFACE)))
     else location_[displaySurface;&Location]:1)
    (SendMessage {displaySurface;&Boundary] 'ToLocate location))

Source File: AIPSTEMP
(TLPlaneSurface
  (DLAMBDA ((planeSurface IndividualConcept (SATISFIES planeSurface df
    PLANESURFACE)))
    (* Edited by J.Gibbons on
     29-Jan-81.)
    (* The location cannot be set
   from outside but it can be
   derived.)
   
   (TLDisplaySurface planeSurface {planeSurface;&Location}:1))
)

(RPAQQ AIPSTEMPTOMAKEFNS (TMDisplaySurface TMRegionNumber))
(DEFINEQ

(TMDisplaySurface
  (LAMBDA (genericDisplaySurface background)
    (* Edited by J.Gibbons on
     19-Dec-80.)
    (Make DISPLAYSURFACE Plane plane Background background Boundary 'Fill)
    )))

(TMRegionNumber
  (DLAMBDA ((planeSurface IndividualConcept (SATISFIES planeSurface df
    PLANESURFACE)))
    (location (LISTP OF FIXP))
    (height FIXP)
    (width FIXP)
    (background FIXP)
    (RETURNS (LISTP OF BMGRegionNumber))
    (* Edited by J.Gibbons on
     2-Feb-81.)
    
    (* Initializes a BMGRegion on the specified BMGPlane according
   to boundary and returns its BMGRegionNumber.)
    
    (<(BMGDefineRegion
    
    )
(<'PLANE {planeSurface;&PlaneNumber}:1 'BACKGROUND background 'MINX location:1 'MINY location:2 'MAXX location:1+width-1 'MAXY location:2+height-1>)

(RPAQQ AIPSTEMPTOSIZEFNS (TSPlaneSurface TSRectangularRegion))

(DEFINEQ

(TSPlaneSurface
[DLAMBDA ((planeSurface IndividualConcept (SATISFIES planeSurface df 'PLANESURFACE)))
  (* Edited by J.Gibbons on 28-Jan-81.)

  (* The size of the PlaneSurface cannot be set from outside but it can be derived from the related BMGRegion.)

  [PROG (height width)
    (if height_{planeSurface;Height}:1 and width_{planeSurface;Width}:1
      then (RETURN <height width>)
    else (RETURN (TSRectangularRegion planeSurface
                   [planeSurface;&Height}:1
                   [planeSurface;&Width}:1)))
]

(TSRectangularRegion
[DLAMBDA ((rectangularRegion IndividualConcept (SATISFIES rectangularRegion df 'RECTANGULARREGION)))
  (height (ONEOF NIL FIXP)))
  (width (ONEOF NIL FIXP)))
  (* Edited by J.Gibbons on 28-Jan-81.)

  (if height or width or !(height_{rectangularRegion;Height}:1
    and width_{rectangularRegion;Width}:1)
    then (if height
      then (SetRoleValues rectangularRegion
        Height@RECTANGULARREGION <height>)
      else height_{rectangularRegion;&Height}:1)
    (if width
      then (SetRoleValues rectangularRegion

Source File: AIPSTEMP
Width@RECTANGULARREGION <width>)
    else width {rectangularRegion; &Width}:1
    (SendMessage {rectangularRegion; &Boundary}:1 'ToSize height width))
    <height width>])
)(DECLARE: DONTEVAL@LOAD EVAL@COMPILEWHEN (NOT (BOUNDP (QUOTE BMGCOMS)))
DONTCOPY
(FILESLOAD <FONTWORK>BMGRECORDSANDVARS..0)
)

(ADDTOVAR CKLONEFILES AIPSTEMP)
STOP
14. SOURCE FILE: AIPSTODRAW

DisplaySurfaceForGroundP ........................................ 27
FindDrawingContextsForDisplay ................................ 28
FindRoleValuesTransitively .................................... 29
FindTopLevelDisplaysOnViewSurface ............................... 30
GetFontCharacterDef ............................................. 31
LDIFFERENCE ...................................................... 32
RotatePointsAroundPoint ........................................ 34
ROUND ............................................................. 33
RoundPoint .......................................................... 35
RoundPoints .......................................................... 36
ScrollFastWindow ................................................. 1
ScrollSlowWindow .................................................. 2
ScrollTTYWindow .................................................... 3
ScrollWindow .......................................................... 37
TDCircle ............................................................. 4
TDCurvec ............................................................. 5
TDDisplay ............................................................. 6
TDDisplayComposite ................................................. 7
TDDisplayItemAbstraction ........................................... 8
TDDisplaySurface ..................................................... 9
TDEdgeSet ........................................................... 10
TDEllipse ............................................................. 11
TDFastWindow ........................................................ 12
TDIFastWindow ....................................................... 13
TDIWNonScrollWindow ............................................... 14
TDISlowWindow ....................................................... 15
TDLineSegment ........................................................ 16
TDNonScrollWindow .................................................. 17
TDPoint ................................................................. 18
TDRectangle ........................................................... 19
TDRregularPolygon ..................................................... 20
TDSlowWindow .......................................................... 21
TDText ................................................................. 22
TDTTYWindow ........................................................... 23
TDWindow ............................................................... 24
TEDisplayAtom ........................................................ 25
TEDisplayComposite ................................................... 26
TransformPoints ........................................................ 38
TransformSize .......................................................... 39
changes to: ScrollFastWindow
previous date: " 2-Feb-81 02:48:19" <NEWAIPS>AIPSTODRAW..62)

(PRETTYCOMPRINT AIPSTODRAWCOMS)

(RPAQQ AIPSTODRAWCOMS (/* The following are the ToDraw functions. They are
called only after all locating has been done. */
(FNS * AIPSTODRAWFNS)
(FNS * AIPSTODRAWERASEFNS)
(DECLARE: COPYWHEN (NOT COMPILEIGNOREDECL)
(RECORDS * AIPSTODRAWRECORDS))
(FNS * AIPSTODRAWUTILITYFNS)
(GLOBALVARS TDRegion)
(DECLARE: DONTEVAL@LOAD EVAL@COMPILWHEN
(NOT (BOUNDP (QUOTE BMGCOMS)))
DONTCOPY
(FILES <FONTWORK>BMGRECORDSANDVARS..0))
(ADDVARS (CKLONEFILES AIPSTODRAW))])

Source File: AIPSTODRAW 156
(ScrollFastWindow
[DLAMBDA ((fastWindow IndividualConcept (SATISFIES fastWindow df
  "FASTWINDOW")
    location)
  (CLISP: MIXED) (* Edited by J.Gibbons on
  2-Feb-81.)

(* Redraws the aperture and its contents according to the new
location or the old substrate location if no new one is
given.)

(PROG ((apertureDisplaySurface {fastWindow;&Aperture;&DisplaySurface}:1)
  (substrate {fastWindow;&Substrate}:1)
  locationOfSubstrate substrateRegion)
  (if location
    then (SetRoleValues fastWindow SubstrateLocation@FASTWINDOW
      <location>)
    else location_ {fastWindow;&SubstrateLocation}:1)
  (locationOfSubstrate {substrate;&Location}:1)
  (substrateRegion_ (TMRegionNumber "HIDDENPLANE"
    <locationOfSubstrate:1+location:1
      locationOfSubstrate:2+location:2>
      {apertureDisplaySurface;&Height}:1
      {apertureDisplaySurface;&Width}:1):1)
  (BMGCopyRegion substrateRegion
    {apertureDisplaySurface;&RegionNumber}:1)
  (BMGFreeRegionNumber substrateRegion)))}}

(ScrollSlowWindow
[DLAMBDA ((slowWindow IndividualConcept (SATISFIES slowWindow df
  "SLOWWINDOW")
    location)
  (CLISP: MIXED) (* Edited by J.Gibbons on
  27-Jan-81.)

(* Redraws the aperture and its contents according to the new
location or the old substrate location if no new one is
given.)

(if location
  then (SetRoleValues slowWindow SubstrateLocation@SLOWWINDOW
    <location>)
  else location_ {slowWindow;&SubstrateLocation}:1)
  (location_ (create PointRecord xCoord_-location:1

157 Source File: AIPSTODRAW
(bind (apertureDisplaySurface (CAR \{slowWindow;&Aperture;&DisplaySurface\}))
first (SendMessage apertureDisplaySurface 'ToDraw) for display
in (FindTopLevelDisplaysOnViewSurface (CAR \{slowWindow;&Substrate\}))
do (SendMessage display 'ToDraw apertureDisplaySurface location)))

(ScrollTTYWindow
[DLAMBDA ((ttyWindow IndividualConcept (SATISFIES ttyWindow df 'TTYWINDOW))
(height (ONEOF NIL FIXP)))
  (* Edited by J.Gibbons on 28-Jan-81.)
  (* Scrolls the TTYWindow up height pixels or, if NIL, a
default amount.)
(BMGScrollRegion height \{ttyWindow;&Substrate;&RegionNumber\};1))

(TDCircle
[DLAMBDA ((circle IndividualConcept (SATISFIES circle df 'CIRCLE))
(displaySurface IndividualConcept (SATISFIES displaySurface df 'DISPLAYSURFACE))
(transformation (LST OF NUMBERP))
(magnification (ONEOF NIL NUMBERP))
(rotationSpec (ONEOF LST NUMBERP))
(displayMode (ONEOF NIL BMGDisplayMode)))
  (* Edited by J.Gibbons on 13-Jan-81.)
  (* Draws a circle.)
(PROG ((regionNumber \{displaySurface;&RegionNumber\};1)
  (center \{circle;&Center\};1)
  (radius \{circle;&Radius\};1))
  (center_ (create PointRecord xCoord_center:1
  yCoord_center:2))
  (center_ (RoundPoint (TransformPoints <center> rotationSpec
  magnification translation):1))
  (radius_ (ROUND (TransformSize radius magnification)))
  (BMGDisplayMode (if displayMode
    else 'ADD)
    regionNumber)
  (BMGMove center: xCoord center: yCoord regionNumber)
  (BMGCircle radius regionNumber)))

Source File: AIPSTODRAW 158
(TDCurve
dl lambda ((curve IndividualConcept (satisfies (curve df 'CURVE)))
(displaySurface IndividualConcept (satisfies displaySurface df
'dISPLAYSURFACE))
(translation (lst of numberp))
(magnification (oneof nil numberp))
(rotationSpec (oneof lst numberp))
(displayMode (oneof nil bmGDisplayMode)))
(* edited by J.Gibbons on
13-Jan-81.)
(* draws a curve via its
approximation)
(prog ((approximation {curve;&approximation}:1))
(sendmessage approximation 'ToDraw displaySurface translation
magnification rotationSpec displayMode))))

(TDDisplay
dl lambda ((display IndividualConcept (satisfies display df 'DISPLAY))
(drawingContext (oneof nil IndividualConcept) (satisfies
'drawingContext
'displayContext df 'DISPLAYSURFACE)
(drawingContext df 'WINDOW)))
(translation (lst of numberp)))
(* edited by J.Gibbons on
19-Jan-81.)
(* draws an individual of
Display when the drawing
context is not specified.)
(if drawingContext df 'DISPLAYSURFACE
then (if (displaySurfaceForGroundP drawingContext
{display;&ground}:1)
then (prog ((location {display;&location}:1)
(size {display;&size}:1)
(orientation {display;&orientation}:1))
(location (_create PointRecord xCoord_location:1
yCoord_location:2))
(location (_TransformPoints <location> NIL NIL
translation):1)
(* draw abstractions.)
(bind abstraction for concept
in (klzGetSuperConcepts display)
when abstraction_(klzGetAbstraction concept)
do (SendMessage abstraction 'ToDraw
drawingContext location size
orientation))
(* Draw DisplayForm
Realizations.)
(bind genericDisplayForm `DISPLAYFORM
for realization in {display;Realization}
when realization df genericDisplayForm
do (SendMessage realization 'ToDraw
drawingContext translation]
(* Draw subDisplays.)
(bind genericDisplay `DISPLAY for realization
in {display;Realization} when realization df genericDisplay
do (SendMessage realization 'ToDraw drawingContext
translation))
elseif drawingContext df `WINDOW
then (SendMessage drawingContext 'ToDrawInWindow display)
elseif `drawingContext
then (for drawingContext in (FindDrawingContextsForDisplay display)
do (SendMessage display 'ToDraw drawingContext)))

{TDDisplayComposite
[DLAMBDA ((displayComposite IndividualConcept (SATISFIES
(displayComposite df
`DISPLAYCOMPOSITE)))
(displaySurface IndividualConcept (SATISFIES displaySurface df
`DISPLAYSURFACE))
(transaction (LST OF NUMBERP))
(magnification (ONEOF NIL NUMBERP))
(rotationSpec (MEMQ NIL 0))
(displayMode (ONEOF NIL BMGDisplayMode)))
(* Edited by J.Gibbons on
16-Jan-81.)

(* Draws an individual of DisplayComposite.
We cannot handle an incoming rotationSpec but this seems to be
the only such limitation.)

(PROG ((components [displayComposite;&Component])
(location [displayComposite;&Location]:1)
(size [displayComposite;&Size]:1)
(orientation [displayComposite;&Orientation]:1))
(location_ (create PointRecord xCoord_location:1
yCoord_location:2))
(location_ (TransformPoints <location> NIL magnification
translation):1)
(size_ (TransformSize size magnification))

Source File: AIPSTODRAW 160
(for component in components
do (SendMessage component 'ToDraw displaySurface location size
orientation displayMode)))}}

{TDDisplayItemAbstraction
[DLAMBDA ((abstraction IndividualConcept (SATISFIES abstraction df
`DISPLAYITEMABSTRACTION))
(displaySurface IndividualConcept (SATISFIES displaySurface df
`DISPLAYSURFACE))
(translation (LST OF NUMBERP))
(magnification (ONEOF NIL NUMBERP))
(rotationSpec (ONEOF NIL NUMBERP)))
(* Edited by J.Gibbons on
16-Jan-81.)*
(* Draws an individual of
DisplayItemAbstraction
according to the parameters
given.)*

(for component in {abstraction;FixedComponent}
do (SendMessage component 'ToDraw displaySurface translation
magnification rotationSpec)))}}

{TDDisplaySurface
[DLAMBDA ((displaySurface IndividualConcept (SATISFIES displaySurface df
`DISPLAYSURFACE))
(baseDisplaySurface (ONEOF NIL IndividualConcept) (SATISFIES
`baseDisplaySurface
or
(baseDisplaySurface df `DISPLAYSURFACE)))
(translation (LST OF NUMBERP))
(magnification (ONEOF NIL NUMBERP))
(rotationSpec (ONEOF NIL NUMBERP))
(displayMode (ONEOF NIL BMGdisplayMode)))
(* Edited by J.Gibbons on
19-Jan-81.)*

(* Fills the associated BMGRegion with its Background
according to displayMode. The excess formal arguments are to
make the function call compatible with other DisplayForms.)*

(PROG ((regionNumber {displaySurface;&RegionNumber}:1))
(BMGDisplayMode (if displayMode

161 Source File: AIPSTODRAW
[10]

(TDEdgeSet

[DLAMBDA ((edgeSet IndividualConcept (SATISFIES edgeSet df 'EDGESET))
(displaySurface IndividualConcept (SATISFIES displaySurface df 'DISPLAYSURFACE))

(translation (LST OF NUMBERP))
(magnification (ONEOF NIL NUMBERP))
(rotationSpec (ONEOF LST NUMBERP))
(displayMode (ONEOF NIL BMGDisplayMode)))

(* Edited by J.Gibbons on 9-Jan-81.)

(* Draws an Edge Set; ie. a set of line segments determined by a list of vertices.)

(PROG ((regionNumber {displaySurface;&RegionNumber} :1)
(vertexList {edgeSet;&VertexList} :1))

(vertexList_ (for vertex in vertexList
collect (create PointRecord xCoord_vertex:1
yCoord_vertex:2)))

(vertexList_ (RoundPoints (TransformPoints vertexList rotationSpec
maginification translation)))

(BMGDisplayMode (if displayMode
else 'ADD)
regionNumber)

(BMGPolyLine vertexList regionNumber)))

{11}

(TDEllipse

[DLAMBDA ((ellipse IndividualConcept (SATISFIES ellipse df 'ELLIPSE))
(displaySurface IndividualConcept (SATISFIES displaySurface df 'DISPLAYSURFACE))

(translation (LST OF NUMBERP))
(magnification (ONEOF NIL NUMBERP))
(rotationSpec (ONEOF LST NUMBERP))
(displayMode (ONEOF NIL BMGDisplayMode)))

(CLISP: MIXED)

(* Edited by J.Gibbons on 13-Jan-81.)

(* Draws an ellipse.)

(PROG ((regionNumber {displaySurface;&RegionNumber} :1))

Source File: AIPSTODRAW 162
(center [ellipse;&Center]:1)
(semiMajorAxis [ellipse;&SemiMajorAxis]:1)
(semiMinorAxis [ellipse;&SemiMinorAxis]:1)
(orientation [ellipse;&Orientation]:1))
(center (create PointRecord xCoord_center:1
         yCoord_center:2))
(center (RoundPoint (TransformPoints <center> rotationSpec
                     magnification translation):1))

(semiMajorAxis_ (ROUND (TransformSize semiMajorAxis magnification))
                  (semiMinorAxis_ (ROUND (TransformSize semiMinorAxis magnification)))
  (if (LISTP rotationSpec)
      then rotationSpec_rotationSpec:1)
      (orientation_ (ROUND (REMAINDER orientation+(if rotationSpec
                       else 0)
                       180))))
  (if orientation gt 90
     then orientation_orientation-180
     elseif orientation lt -90
     then orientation_orientation+180)
  (BMGDisplayMode (if displayMode
                    else 'ADD)
                    regionNumber)
  (BMGMove center:xCoord center:yCoord regionNumber)
  (BMGEllipse semiMinorAxis semiMajorAxis orientation regionNumber))

[TDFastWindow
DLAMBDA ((fastWindow IndividualConcept (SATISFIES (fastWindow df
         drawingContext) 'FASTWINDOW)))
         (* Edited by J.Gibbons on
         2-Feb-81.)
         (* Draws a FastWindow in the specified drawing context.
It must locate the substrate on the HIDDENPLANE, draw the
appropriate displays there, and then copy the substrate to the
aperture.)

  (if drawingContext df 'DISPLAYSURFACE
     then (PROG ((substrate [fastWindow;&Substrate]:1))
               (TDWindow fastWindow drawingContext)
               (SendMessage substrate 'ToLocate
                (FindRegion "HIDDENPLANE
                [substrate;&Height]:1

163 Source File: AIPSTODRAW
(SendMessage substrate 'ToDraw 'HIDDENPLANE)
(for display in (FindTopLevelDisplaysOnViewSurface substrate)
    do (SendMessage display 'ToDraw substrate))
(SendMessage fastWindow 'Scroll))
elseif drawingContext df 'WINDOW
    then (SHOULDNT)
elseif ~drawingContext
    then (for drawingContext in (FindDrawingContextsForDisplay fastWindow)
        do (SendMessage fastWindow 'ToDraw drawingContext)))

(TDIWFastWindow
[DLAMBDA ((fastWindow IndividualConcept (SATISFIES fastWindow df 'FASTWINDOW))
    (display IndividualConcept (SATISFIES display df 'DISPLAY)))
    (* Edited by J.Gibbons on 2-Feb-81.)
(SendMessage display 'ToDraw {fastWindow;&Substrate}:1)
(SendMessage fastWindow 'Scroll))]

(TDIWNonScrollWindow
[DLAMBDA ((nonScrollWindow IndividualConcept (SATISFIES nonScrollWindow df 'NONSCROLLWINDOW))
    (display IndividualConcept (SATISFIES display df 'DISPLAY)))
    (* Edited by J.Gibbons on 27-Jan-81.)
    (* Knows how to draw a Display in a NonScrollWindow.)
(SendMessage display 'ToDraw {nonScrollWindow;&Substrate}:1))]

(TDIWSlowWindow
[DLAMBDA ((slowWindow IndividualConcept (SATISFIES slowWindow df 'SLOWWINDOW))
    (display IndividualConcept (SATISFIES display df 'DISPLAY)))
    (CLISP: MIXED)
    (* Edited by J.Gibbons on 27-Jan-81.)
    (* Knows how to draw Displays in SlowWindows.)
    (PROG ((substrateLocation {slowWindow;&SubstrateLocation}:1))

Source File: AIPSTODRAW 164
(SendMessage display 'ToDraw
  [slowWindow;&Aperture;&DisplaySurface];1
  (create PointRecord xCoord_-substrateLocation:1
       yCoord_-substrateLocation:2))])}

(TDLineSegment
 [DLAMBDA ((lineSegment IndividualConcept (SATISFIES lineSegment df
            'LINESEGMENT))
            (displaySurface IndividualConcept (SATISFIES displaySurface df
            'DISPLAYSURFACE))
            (translation (LST OF NUMBERP))
            (magnification (ONEOF NIL NUMBERP))
            (rotationSpec (ONEOF LST NUMBERP))
            (displayMode (ONEOF NIL BMGDisplayMode)))
  (* Edited by J.Gibbons on 9-Jan-81.)
  (* Draws a line segment.)
  (PROG ((regionNumber [displaySurface;&RegionNumber]:1)
      (endPoints [lineSegment;&EndPoint]))
    (endPoints_(for point in endPoints
      collect (create PointRecord xCoord_point:1
               yCoord_point:2)))
    (endPoints_(RoundPoints (TransformPoints endPoints rotationSpec
                            magnification translation)))
    (BMGDisplayMode (if displayMode
                      else 'ADD)
                     regionNumber)
    (BMGLine endPoints:1:xCoord endPoints:1:yCoord endPoints:2:xCoord
             endPoints:2:yCoord regionNumber)))

(TDNonScrollWindow
 [DLAMBDA ((nonScrollWindow IndividualConcept (SATISFIES nonScrollWindow df
            'NONSCROLLWINDOW))
            drawingContext)
  (* Edited by J.Gibbons on 27-Jan-81.)
  (* Draws a NonScrollWindow in the specified drawing context.)
  (if drawingContext df 'DISPLAYSURFACE
      then (PROG ((substrate [nonScrollWindow;&Substrate]:1))
        (TDWindow nonScrollWindow drawingContext)
        (for display in (FindTopLevelDisplaysOnViewSurface substrate)
          do (SendMessage display 'ToDraw substrate)))

Source File: AIPSTODRAW
elseif drawingContext df 'WINDOW
then (SHOULDNT)
elseif ~drawingContext
then (for drawingContext in (FindDrawingContextsForDisplay
nonScrollWindow)
do (SendMessage nonScrollWindow 'ToDraw drawingContext))])

{18}

(TDPoint
[DLAMBDA ((point IndividualConcept (SATISFIES point df 'POINT))
(displaySurface IndividualConcept (SATISFIES displaySurface df
'DISPLAYSURFACE))

(translation (LST OF NUMBERP))
(magnification (ONEOF NIL NUMBERP))
(rotationSpec (ONEOF LST NUMBERP))
(displayMode (ONEOF NIL BMGDisplayMode)))

(* Edited by J.Gibbons on 9-Jan-81.)

(* Draws a point.)

(PROG ((regionNumber [displaySurface;&RegionNumber]:1)
(location {point;&Location}:1))
(location_ (create PointRecord xCoord 'Tlocation:l
yCoord location: 2))
(location_ (RoundPoint (TransformPoints <location> rotationSpec
magnification translation)
:1))

(BMGDisplayMode (if displayMode
else 'ADD)
regionNumber)
(BMGPoint location:xCoord location:yCoord regionNumber)))

{19}

(TDRectangle
[DLAMBDA ((rectangle IndividualConcept (SATISFIES rectangle df 'RECTANGLE))
(displaySurface IndividualConcept (SATISFIES displaySurface df
'DISPLAYSURFACE))

(translation (LST OF NUMBERP))
(magnificationSpec (ONEOF NIL NUMBERP))
(rotationSpec (ONEOF LST NUMBERP))
(displayMode (ONEOF NIL BMGDisplayMode))

(CLISP: MIXED)

(* Edited by J.Gibbons on 9-Jan-81.)

(* Draws a rectangle from the lowerLeft-upperRight
representation. Orientation is with respect to the center of
the rectangle.)

Source File: AIPSTODRAW 166
(PROG ((regionNumber {displaySurface;&RegionNumber}:1)
  (lowerLeft [rectangle;&LowerLeft]:1)
  (upperRight [rectangle;&UpperRight]:1)
  (orientation [rectangle;Orientation]:1)
  corners)
  (lowerLeft_ (create PointRecord xCoord lowerLeft:1
                 yCoord lowerLeft:2))
  (upperRight_ (create PointRecord xCoord upperRight:1
                  yCoord upperRight:2))
  (corners_ (RoundPoints (TransformPoints
    (TransformPoints
      <(create PointRecord
         xCoord_lowerLeft:xCoord
         yCoord_upperRight:yCoord)
      upperRight
      <(create PointRecord
         xCoord_upperRight:xCoord
         yCoord_lowerLeft:yCoord)
      lowerLeft>
      <orientation (create
        PointRecord xCoord他们会
        lowerLeft:xCoord+upperRight:xCoord)
        /2.0 yCoord-(
        lowerLeft:yCoord+upperRight:yCoord)
        /2.0)
      >)
      rotationSpec magnification translation)))
  (BMGDisplayMode (if displayMode
                   else 'ADD)
                   regionNumber)
  (BMGMove corners:4:xCoord corners:4:yCoord regionNumber)
  (BMGPolyLine corners regionNumber))))

{20}

(TDRegularPolygon
[DLAMBDA ((regularPolygon IndividualConcept (SATISFIES
  (regularPolygon df
 `REGULARPOLYGON))))
  (CLISP: MIXED)
  (* Edited by J.Gibbons on
  31-Jan-81.)
  (* Draws a regular polygon
  knowing its Center,
  Orientation, Radius, and
  Order.)

[PROG ((center [regularPolygon;Center]:1)
   (radius [regularPolygon;Radius]:1)
   (order [regularPolygon;Order]:1)
(orientation (OR (regularPolygon:Orientation) 1 0))
vertex vertexList)
(if center=NIL or radius=NIL or order=NIL
then (* insufficient information for drawing)
(TDFail 'TDRegularPolygon regularPolygon)
else [vertexList_ (bind (iterationAngle_360./order)
(listOfPoint <<center:1+radius
center:2>>)]
from 1 to order as angle from orientation
by iterationAngle
collect (vertex_ (TransformPoint
(RotatePointsAroundPoint
< center angle > listOfPoint)
:1])
(BMGMove vertex:1 vertex:2)
(BMGPolyLine (for vertex in vertexList
collect <vertex:1 ! vertex:2>)]))

[TDSlowWindow
[DLAMBDA ((slowWindow IndividualConcept (SATISFIES slowWindow df 'SLOWWINDOW))
drawingContext)
(* Edited by J.Gibbons on 26-Jan-81.)
(* Draws a SlowWindow on the specified DisplaySurface.)
(if drawingContext df 'DISPLAYSURFACE
then (TDWindow slowWindow drawingContext)
(SendMessage slowWindow 'Scroll)
elseif drawingContext df 'WINDOW
then (SHOULDNT)
elseif 'drawingContext
then (for drawingContext in (FindDrawingContextsForDisplay slowWindow)
do (SendMessage slowWindow 'ToDraw drawingContext))))]

[TDTTYWindow
[DLAMBDA ((ttyWindow IndividualConcept (SATISFIES ttyWindow df 'TTYWINDOW))
drawingContext)
(* Edited by J.Gibbons on 27-Jan-81.)
(* Draws a TTYWindow and ties the BMGTTYRegion to its...
(if drawingContext df 'DISPLAYSURFACE
  then (TDDWindow ttyWindow drawingContext)
  (BMGTTYRegion {ttyWindow;&Substrate;&RegionNumberl :1)
  (BMGResetTTY)
  elseif drawingContext df 'WINDOW
    then (SHOULDNT)
  elseif "drawingContext
    then (for drawingContext in (FindDrawingContextsForDisplay
      ttyWindow)
      do (SendMessage ttyWindow 'ToDraw drawingContext)))

(* Edited by J.Gibbons on 31-Jan-81.)

(* Draws the specified text string at the desired location
  using the correct font.)

[PROG ((fontNumber {text;&Font} :1)
  (lineSpacing -Itext;&LineSpacing} :1)
  (location {text;&Location}:1)
  (string {text;&String}:1)) (* Set the Graphics region and
  its font and starting
  location.)

  (BMGRegion (text;&Ground;&RegionNumber}:1)
  (BMGFont fontNumber)
  (BMGMove location:1 location:2+[text;&Height}:1-(BMGDescribeFont
    fontNumber):LoadedFont\FontRecord:Font\Baseline
  (* Output the string.)

    (bind [breakChars _ (CONSTANT (MAKEBITTABLE '([10 13 31]
        lineBeginPos-1
        subString_"="breakChar breakPos
        while breakPos_(STRPOSL breakChars string lineBeginPos)
        do (if subString_(SUBSTRING string lineBeginPos breakPos-1
          subString)
           then (BMGString subString)
           else subString_"

            (lineBeginPos_breakPos+1)
            (if breakChar_(CHCON1 (NTHCHAR string breakPos)=31

169Source File: AIPSTODRAW
then

(* LISP's internal end of line character. Treat it like a carriage return, line feed.)*

(BMGMove location:1 ((BMGY)+lineSpacing))
elseif breakChar=10
then (* Bare line feed.)*
(BMGMoveRel (MakeTextWidth subString fontNumber):1 lineSpacing)
elseif (CHCons (NTHCHAR string lineBeginPos))=10
then (* Carriage return, line feed.)*
(lineBeginPos lineBeginPos+1)
(BMGMove location:1 ((BMGY)+lineSpacing))
else (* Bare carriage return.)*
(BMGMove location:1 (BMGY))
finally (if subString_=(SUBSTRING string lineBeginPos NIL subString)
then (BMGString subString)])

[24]

(TDWindow [DLAMBDA ((window IndividualConcept (Satisfies (window df `WINDOW)))
(drawingContext (ONEOP NIL IndividualConcept) (Satisfies
(OR
`drawingContext
(drawingContext df `DISPLAYSURFACE)
(drawingContext df `WINDOW)))
(translation (LST OF NUMBERP))) (Edited by J.Gibbons on 19-Jan-81.)

(* Draws the Window as a Display on the specified DisplaySurface. The parts must be drawn in the proper order; eg the DisplaySurface which provides the background for the window as a whole must be drawn first.)*

(if drawingContext df `DISPLAYSURFACE
then (if (DisplaySurfaceForGroundP drawingContext window;&Context):1)
then (bind drawingSurface (window;&DisplaySurface):1
first (SendMessage drawingSurface 'ToDraw)
for realization in (window;Realization)
unless realization=drawingSurface)
do (SendMessage realization 'ToDraw drawingContext translation))

elseif drawingContext df 'WINDOW
  then (* we cannot handle this currently)

  (SHOULDNT)

elseif ~drawingContext
  then (for drawingContext in (FindDrawingContextsForDisplay window)
      do (SendMessage window 'ToDraw drawingContext))))]

(RPAQQ AIPSTODRAWERASEFNS (TEDisplayAtom TEDisplayComposite))

(DEFINEQ

(TEDisplayAtom
  [LAMBDA (displayAtom displaySurface translation magnification rotationSpec)
   (* Edited by J.Gibbons on 13-Jan-81.)

   (* Erases the displayAtom by drawing it in FLIP mode.
      This works satisfactorily for most DisplayAtoms.
      For those where it does not, there will be more specialized erasing functions.)

   (SendMessage displayAtom 'ToDraw displaySurface translation magnification rotationSpec 'FLIP))

{25}

(TEDisplayComposite
  [DLAMBDA ((displayComposite IndividualConcept (SATISFIES (displayComposite df 'DISPLAYCOMPOSITE)))
         (displaySurface IndividualConcept (SATISFIES displaySurface df 'DISPLAYSURFACE))
         (translation (LST OF NUMBERP))
         (magnification (ONEOF NIL NUMBERP))
         (rotationSpec (MEMQ NIL 0)))
   (* Edited by J.Gibbons on 13-Jan-81.)

   (* Erases an individual of DisplayComposite.
      We cannot handle an incoming rotationSpec but this seems to be the only such limitation. This is identical to TDDisplayComposite except that the recursive request is to erase rather than to draw.)

171 Source File: AIPSTODRAW
(PROG ((components {displayComposite;&Component})
   (location {displayComposite;&Location}:1)
   (size {displayComposite;&Size}:1)
   (orientation {displayComposite;&Orientation}:1))
   (location (TransformPoints <location> NIL magnification translation):1)
   (size (TransformSize size magnification))
   (for component in components
      do (SendMessage component 'ToErase displaySurface location size orientation))))

(DECLARE: COPYWHEN (NOT COMPILEIGNOREDECL))

(RPAQQ AIPSTODRAWRECORDS (PointRecord))
(DECLARE: EVAL@COMPILE

(RECORD PointRecord (xCoord . yCoord)
   xCoord _ 0 yCoord _ 0)
)

(RPAQQ AIPSTODRAWUTILITYFNS (DisplaySurfaceForGroundP
   FindDrawingContextsForDisplay
   FindRoleValuesTransitively
   FindTopLevelDisplaysOnViewSurface
   GetFontCharacterDef
   LDIFFERENCE ROUND
   RotatePointsAroundPoint
   RoundPoint
   RoundPoints
   ScrollWindow
   TransformPoints
   TransformSize)

(DEFINEQ

{27}

(DisplaySurfaceForGroundP
 [LAMBDA (displaySurface ground) (* Edited by J. Gibbons on 19-Jan-81.)
   (OR ground=displaySurface (AND ~(ground df 'DISPLAYSURFACE)
      (for window in {ground;Window}
         thereis (CAR
            [window;Aperture;DisplaySurface])
        =displaySurface))
)

{28}

Source File: AIPSTODRAW 172
(FindDrawingContextsForDisplay
[DLAMBDA ((display IndividualConcept (SATISFIES display df `DISPLAY)))
  (* Edited by J.Gibbons on 28-Jan-81.)
  (* Returns a list of the relevant drawing contexts for the Display and all of its subDisplays.)

(bind ground_{display;&Groundl:l
genericDisplay_`DISPLAY
drawingContexts windows for realization
in {display;Realization} when realization df genericDisplay
join (FindDrawingContextsForDisplay realization)
finally (RETURN (INTERSECTION drawingContexts_
<
  !(if windows_{ground;Window}
    then T windows>
    elseif ground df `DISPLAYSURFACE
    then <ground>)
    ! $$VAL>
drawingContexts)))]])

{29}

(FindRoleValuesTransitively
[DLAMBDA ((individualConcept IndividualConcept)
  (genericRole Role)
  (RETURNS (LST OF RoleValue)))
  (* Edited by J.Gibbons on 27-Aug-80.)
  (* Finds all of the individuals which are transitively related to the individualConcept by virtue of being a Role Filler of the genericRole. For example, if concept FOO has generic role BAR and individuaters A, B, and C such that B fills the BAR role of A and C fills the BAR role B, then both B and C are returned for A and BAR.)

(PROG (fillerLst)
  (RETURN < !!(fillerLst_ (KLFindRoleValues individualConcept
    genericRole))
    !(for filler in fillerLst
      join (FindRoleValuesTransitively filler genericRole)
    >))))))

173 Source File: AIPSTODRAW
(FindTopLevelDisplaysOnViewSurface
[DLAMBDA ((viewSurface IndividualConcept (SATISFIES (viewSurface `VIEWSURFACE)))
  (RETURNS (LST OF IndividualConcept)))
(* Edited by J.Gibbons on 29-Jul-80.)

(* This returns a list of Displays each of which 1: has its Ground role filled by viewSurface and 2: does not fill the Realization role of some other Display having viewSurface as Ground.)

[PROG (displays)
  (displays_ (bind (groundRole _ (KLFindOneNamedGenericRole `DISPLAY `Ground))
    for groundlRole in (KLGetValueDescriptionInverses viewSurface)
    collect (KLGetConceptOfRole groundlRole)
    when (KLIsRoleDescendantP groundlRole groundRole))
  (RETURN (LDIFFERENCE displays
    (for display in displays
      join (FindRoleValuesTransitively display
        (KLFindOneNamedGenericRole `DISPLAY `Realization)))))

(GetFontCharacterDef
[DLAMBDA ((fontNumber SMALLP (SATISFIES fontNumber ge 0
  and fontNumber le BMGMaxNumberOfFonts))
  (characterCode SMALLP (SATISFIES characterCode ge 0
    and characterCode le 127)))
  (* Edited by J.Gibbons on 18-Sep-80.)
  (* Returns the BMGCharDefRecord for the specified characterCode in the specified font.)
  'ELT (BMGDescribeFont fontNumber):LoadedFont\FontRecord:Font\CharArray 1+characterCode))}

(LDIFFERENCE
[DLAMBDA (LIST1 LIST2) (* Edited by J.Gibbons on
  (* Edited by J.Gibbons on 711A!) 174

--- File: AIPSTODRAW
Report No. 4752

Bolt Beranek and Newman Inc.

31-Jan-81.

(* WE NEED THIS SINCE IT IS UNDEFINED IN THE SMALL CKLONE)

(for X in LIST1 collect X unless X mem LIST2))

[33]

(ROUND

[DLAMBDAX NUMBERP])

(* Edited by J.Gibbons on 7-Jan-81.)

(if (FIXP x)
    then x
  else (FIX (if (MINUSP x)
             then x-.5
             else x+.5)))]

[34]

(RotatePointsAroundPoint

[DLAMBDAX LST OF (LISTP OF NUMBERP))
  (rotationSpec ONEOF NUMBERP LST))]}

(CLISP: MIXED) (* Edited by J.Gibbons on 8-Jan-81.)

(* Rotates a list of points counterclockwise from the positive xAxis around a center point according to orientation. Assumes a cartesian coordinate system.)

(if rotationSpec and pointsLst
    then (PROG ((theta (if (NUMBERP rotationSpec)
                          then rotationSpec
                        else rotationSpec:1))
         [center (if (LISTP rotationSpec)
                      then rotationSpec:2
                    else (CONSTANT (create PointRecord]
                      cos sin xConstant yConstant)
         (RETURN (if theta and -(EQP theta 0)
                     then (cos (COS theta))
                   (sin (SIN theta))
                   (xConstant center:xCoord*(1.0-cos)
                    +center:yCoord*sin)
                   (yConstant center:yCoord*(1.0-cos)
                    -center:xCoord*sin)
                   (for point in pointsLst
collect (create PointRecord xCoord_
                 xConstant+point:xCoord*cos-point:yCoord*sin_
                 yCoord_

175 Source File: AIPSTODRAW
yConstant+point:xCoord*sin+point:yCoord*cos))
else pointsLst)))}

(RoundPoint
[DLAMBDA (point)
(* Edited by J.Gibbons on 8-Jan-81.)
(* ROUNDS the coordinates of a point if either of them is a FLOATP)

(if (FLOATP point:xCoord) or (FLOATP point:yCoord)
then (create PointRecord
  xCoord _ (ROUND point:xCoord)
yCoord _ (ROUND point:yCoord)
else point])}

(RoundPoints
[DLAMBDA ((pointsLst (LST OF (LISTP OF NUMBERP))))
(* Edited by J.Gibbons on 9-Jan-81.)
(* Round all of the coordinates of points in pointsLst to integers.)

(for point in pointsLst collect (RoundPoint point))))

(ScrollWindow
[DLAMBDA ((scrollWindow IndividualConcept (SATISFIES (scrollWindow df "SCROLLWINDOW")))
  (substrateLocation LISTP))
(* Edited by J.Gibbons on 1-Aug-80.)
(* Sets the Substrate Location of the ScrollWindow and redraws the Aperture.)

(KLChangeRoleValue (KLFindOneNamedInstanceRole scrollWindow 'SubstrateLocation)
  <substrateLocation>)
(APPLY* (KLFindIData scrollWindow '(ToDrawAperture)):1 scrollWindow)])
(TransformPoints
  [DLAMBDA ((pointsLst (LST OF (LISTP OF NUMBERP)))
    (rotationSpec (ONEOF LST NUMBERP))
    (magnification (ONEOF NIL NUMBERP))
    (translation (ONEOF NIL (LISTP OF NUMBERP))))
  (CLISP: MIXED))

(* Edited by J.Gibbons on 13-Jan-81.)
(* Provides a uniform transformation mechanism for the ToDraw functions.)

(if magnification
  else magnification_1)
(if translation
  else translation_1 (CONSTANT (create PointRecord)))
(for point in (RotatePointsAroundPoint pointsLst rotationSpec)
  collect (create PointRecord xCoord_translation:
xCoord + magnification*point:xCoord
  yCoord_translation:yCoord + magnification*point:yCoord)))

(TransformSize
  [DLAMBDA ((size NUMBERP)
    (magnification (ONEOF NIL NUMBERP)))
  (CLISP: MIXED))

(* Edited by J.Gibbons on 13-Jan-81.)
(* Transforms the size parameter by magnification.)

(if magnification
  then magnification*size
  else size))
)
(DECLARE: DNEW@COMPILE DONTCOPY
(ADDTOVAR GLOBALVARS TDRegion)
)
(DECLARE: DONTEVAL@LOAD EVAL@COMPILEWHEN (NOT (BOUNDP (QUOTE BMGCOMS)))
DONTCOPY
.FILESLOAD <FONTWORK>BMGRECORDSANDVARS..0)
)
(ADDTOVAR CKLONEFILES AIPSTODRAW)
STOP
15. SOURCE FILE: AIPSTS

InitTIMESYSTEM..............1
InitTimeSystemConcepts......2
InitUT........................3
(FILECREATED "10-Jul-80 23:58:18" <NEWAIPS>AIPSTS..2 1129
   changes to: AIPSTSCOMS
   previous date: "9-Jul-80 14:54:20" <NEWAIPS>AIPSTS..1)

(PRETTYCOMPRINT AIPSTSCOMS)

(RPAQQ AIPSTSCOMS ((FNS * AIPSTSINITPNS)
   (ADDVARS (CKLONEFILES AIPSTS))))

(RPAQQ AIPSTSINITPNS (InitTIMESYSTEM InitTimeSystemConcepts InitUT))
(DEFINEQ

   {1}

(InitTIMESYSTEM
 [LAMBDA NIL
   [concept TIMESYSTEM
      (* A coordinate system for describing events.)
      (specializes COORDINATESYSTEM)
      [roleset NIL
       (mods Dimensionality@COORDINATESYSTEM)
       (vr 1)])]]

   {2}

(InitTimeSystemConcepts
 [LAMBDA NIL
   (* Initializes concepts having to do with describing time intervals and the timings of events.)
   (InitTIMESYSTEM))

   {3}

(InitUT
 [LAMBDA NIL
   (* Edited by F.Zdybel on 9-Jul-80.)
   [concept UT
    (specializes TIMESYSTEM)
    [roleset NIL

   Source File: AIPSTS 180
(mods Name@CoordinateSystem)
  (vr "Universal Time")]]])
)

(ADDTOVAR CKLONEFILES AIPSTS)
STOP
16. SOURCE FILE: AIPSUTILITY

SendMessage....1
SetRoleValues..2
changes to: SendMessage

previous date: "16-Dec-80 23:54:51" <NEWAIPS>AIPSUTILITY..8)

(PRETTYCOMPRINT AIPSUTILITYCOMS)

(RPAQQ AIPSUTILITYCOMS [(FNS * AIPSUTILITYFNS)
  (VARS (SENDMESSAGENOBREAKFLG)
   (SENDMESSAGENOERRMESSFLG))
  (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
   COMPILERVARS (ADDVARS (NLAMA)
   (NLAML)
   (LAMA SendMessage))]

(RPAQQ AIPSUTILITYFNS (SendMessage SetRoleValues))

(DEFINEQ

(SendMessage
  [LAMBDA argList
t
  (* Edited by J.Gibbons on 19-Dec-80.)

(* Attached procedures on IndividualConcepts are invoked via
 SendMessage (individualConcept procedureTag . args).
 Procedures inherited by individualConcept according to the
 iTag procedureTag are sequentially executed until one
 succeeds, in which case SendMessage passes back the value
 returned by the inherited procedure. If none succeed, an ERROR
 is generated which can be controlled by the variables
 SENDMESSAGENOBREAKFLG and SENDMESSAGENOERRMESSFLG.
 An inherited procedure is APPLY'd to the same argument list as
 SendMessage except that procedureTag is removed.)

(bind value (procedureArgList _ (for index to argList unless index=2
  collect (ARG argList index)))
  for attachedProcedure in (KLFindIData (the Concept (ARG argList 1))
   (MKLIST (ARG argList 2)))
  thereis (NLSETQ value_ (APPLY attachedProcedure procedureArgList))
  finally (if $$VAL
    then (RETURN value)
    else (ERROR (ARG argList 1)
      (CONCAT "doesn't know how to process ")

Source File: AIPSUTILITY 184
(ARG argList 2))
SENDMESSAGENOBREAKFLG SENDMESSAGENOERRMESSFLG)

(SetRoleValues
[DLAMBDA ((iConcept IndividualConcept)
  (parentRole RoleSet)
  (roleValues (LST OF RoleValue)))
  (* Edited by J.Gibbons on 16-Dec-80.)

  (* This function establishes a new set of role fillers for the
   parentRole of the iConcept. It changes role values of existing
   instance roles, deletes excess instance roles if any, or adds
   instance roles if necessary.)

  (bind (instanceRoles _ (KLFindInstanceRoles iConcept parentRole))
    for roleValue in roleValues do (if instanceRoles
    then (KLC ChangeRoleValue
       instanceRoles:1
      <roleValue>)

      instanceRoles instanceRoles:1
    else (KLSatisfyRole parentRole
      iConcept
      <roleValue>))

    finally (for instanceRole in instanceRoles
    do (KLR eMoveRole instanceRole))))
)

(RPAQ SENDMESSAGENOBREAKFLG NIL)
(RPAQ SENDMESSAGENOERRMESSFLG NIL)
(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA SendMessage)
)
STOP
17. SOURCE FILE: AIPSWINDOW

InitAPERTURE........1
InitFASTWINDOW.......2
InitNONSCROLLWINDOW..3
InitSCROLLWINDOW.....4
InitSLOWWINDOW.......5
InitTTYWINDOW........6
InitWINDOW...........7
InitWindowConcepts...8
OpenWindow............9
TLAperture...........10
TLWindow...............11
TMAperture...........12
TMFastWindow........13
TMNonScrollWindow....14
TMSlowWindow..........15
TMTTYWindow..........16
TSAperture...........17
TSNonScrollWindow...18
TSWindow............19
(FILECREATED "2-Feb-81 01:15:28" <NEWAIPS>AIPSWINDOW..122 18798)

changes to: TMFastWindow

previous date: "2-Feb-81 01:00:00" <NEWAIPS>AIPSWINDOW..121)

(PRETTYCOMPRINT AIPSWINDOWCOMS)

(RPAQQ AIPSWINDOWCOMS ((FNS * AIPSWINDOWINITFNS)
  (FNS * AIPSWINDOWUTILITYFNS)
  (FNS * AIPSWINDOWTOLOCATEFNS)
  (FNS * AIPSWINDOWTOMAKEFNS)
  (FNS * AIPSWINDOWTOSIZEFNS)
  (GLOBALVARS AIPSWUSERMODEL)
  (ADDVARS (CKLONEFILES AIPSWINDOW))))

(RPAQQ AIPSWINDOWINITFNS (InitAPERTURE InitFASTWINDOW InitNONSCROLLWINDOW
  InitSCROLLWINDOW InitSLOWWINDOW
  InitTTYWINDOW InitWINDOW
  InitWindowConcepts))

(DEFINEQ

(InitAPERTURE
  [LAMBDA NIL (* Edited by J.Gibbons on 30-Dec-80.)

  [concept APERTURE
    (specializes DISPLAYCOMPOSITE)
    [roleset Border
      (diffs Component#DISPLAYCOMPOSITE)
      (modality Obligatory)
      (vr RECTANGLE)
      (derivation (TMRectangle))]
    [roleset DisplaySurface
      (diffs Component#DISPLAYCOMPOSITE)
      (modality Obligatory)
      (vr DISPLAYSURFACE)]
    (itags (ToLocate TLAperture)
      (ToMake TMAperture)
      (ToSize TSAperture)))]

(InitFASTWINDOW)

Source File: AIPSWINDOW 188
[LAMBDA NIL (* Edited by J.Gibbons on 28-Jan-81.)]

[concept FASTWINDOW
(specializes SCROLLWINDOW)
roleset NIL
(mods Substrate@SCROLLWINDOW)
(vr INVISIBLESURFACE)
(derivation (SeekInvisibleSubstrate
(Prerequisites Application)
(Arguments [{$$CONCEPT;Application}:1])
;Ground):1))]

(* Any Window whose substrate is an actual chunk of map memory can scroll very quickly via BMGCpCopyRegion. Hence the name FastWindow.)

(itags (Scroll ScrollFastWindow)
(ToDraw TDFastWindow)
(ToDrawInWindow TDIWFastWindow)
(ToMake TMFastWindow))]

(InitNONSCROLLWINDOW
[LAMBDA NIL (* Edited by J.Gibbons on 28-Jan-81.)

[concept NONSCROLLWINDOW
(specializes WINDOW)

(* The only distinguishing structural characteristic of NonScrollWindows at this time is that their Substrates are identical with the DisplaySurfaces of their Apertures.)

[roleset NIL
(mods Aperture@WINDOW)
(derivation (TMAperture (Prerequisites Substrate)
(Arguments NIL {$$CONCEPT;Substrate}:1)))]

(itags (ToDraw TDNOnNonScrollWindow)
(ToDrawInWindow TDINNonScrollWindow)
(ToMake TMMNonScrollWindow)
(ToSize TSNonScrollWindow))]

[4]
(InitSCROLLWINDOW
[LAMBDA NIL (* Edited by J. Gibbons on 23-Nov-80.)

[concept SCROLLWINDOW
(specializes WINDOW)
[roleset ScrollBar
  (diffs Realization@DISPLAY)
  (number (0 2))
  (vr CONTROLBAR)]

(* Designations on the Scroll Bar of a Window can be used to modify the description filling the Window's Location Role.)

[roleset SubstrateLocation
  (vr LISTP)
  (defaultfiller '(0 0))]

(* The location of the Window's origin (the origin of its display region) on the coordinate system of the Substrate.)

(preferredprototype "FASTWINDOW]]])

[5]

(InitSLOWWINDOW
[LAMBDA NIL (* Edited by J. Gibbons on 20-Jan-81.)

[concept SLOWWINDOW

(* A Window which scales and scrolls by causing the stuff being viewed through it to be re-presented.)

(specializes SCROLLWINDOW)
[roleset ScaleBar
  (diffs Realization@DISPLAY)
  (vr CONTROLBAR)]

(* Designations in the Scale Bar of a Window can be used to modify the descriptions filling the Entry and Exit Roles of the CoordSys of Window's Substrate (i.e., the scale of the projection through the Window). Since there is no hardware level stuff for doing scaling, it must be accomplished by

Source File: AIPSWINDOW 190
re-presentation, hence can only be accomplished by SlowWindows.)

(ittags (Scroll ScrollSlowWindow)
 (ToDraw TDSlowWindow)
 (ToDrawInWindow TDIWSlowWindow)
 (ToMake TMISlowWindow))])

[InitTTYWINDOW
 [LAMBDA NIL
 (* Edited by J.Gibbons on 27-Jan-81.)

 [concept TTYWINDOW

 (* Individuals of TTYWindow provide the Substrates for the BMGTTYRegion, ie. the destination of system TTY output.)

 (specializes SCROLLWINDOW)
 [roleset NIL
 (mods Substrate@SCROLLWINDOW)
 (derivation (Copy (Prerequisites Aperture)
 (Arguments {$$CONCEPT;Aperture;&DisplaySurface}:1)
 ))

 (ittags (Scroll ScrollTTYWindow)
 (ToDraw TDTTYWindow)
 (ToMake TMTTYWindow))])

[InitWINDOW
 [LAMBDA NIL (* Edited by J.Gibbons on 27-Jan-81.)

 [concept WINDOW

 (* Window is a concept that serves many purposes.
 In the first place, a Window is a convenient way to organize and mobilize the use of the limited available display area.
 In the second place, it can serve as a context for the interpretation of input. Finally, a Window constitutes an important part of the environment for the realization of a display. This explains how it is possible to implement some scrolling in a way that eliminates undesirable edge effects: the "position" of the Window over the substrate can influence the re-layout and re-realization of the underlying...
Presentations.)
(specializes DISPLAY)

(* The Context is the VisibleSurface on which the Window gets displayed. It must either be specified or defaulted.)

[roleset Context
  (mods Ground@DISPLAY)
  (vr DISPLAYSURFACE)
  (defaultfiller "CONRACSCREEN")
]

(* The Substrate is the ViewSurface on which the Window opens. Currently there can be only one. Perhaps in the future we will allow more in which case the images will be superimposed. The Substrate must be specified.)

[roleset Substrate
  (modality Obligatory)
  (vr VIEWSURFACE)
]

(* The Aperture is the VisibleSurface on which the portion of the Substrate covered by the Window is displayed.)

[roleset Aperture
  (diffs Realization@DISPLAY)
  (modality Obligatory)
  (vr APERTURE)
  (derivation (TMAperture (Prerequisites Context)
    (Arguments NIL {$$CONCEPT;Context;&Plane}:1 T))))

(* The Border is a Rectangle which also acts as the Boundary of the Window's DisplaySurface. It gets drawn to pictorially delimit the extent of the Window.)

[roleset Border
  (diffs Realization@DISPLAY)
  (vr RECTANGLE)
  (modality Obligatory)
  (derivation (Copy (Prerequisites DisplaySurface)
    (Arguments {$$CONCEPT;DisplaySurface;&Boundary}:1))}

Source File: AIPSWINDOW
(* The DisplaySurface is all and only the display region on
which the Window and all of its parts are drawn.)

[roleset DisplaySurface
  (diffs Realization@DISPLAY)
  (vr DISPLAYSURFACE)
  (modality Obligatory)
  (derivation (TMDisplaySurface (Prerequisites Context)
                   (Arguments NIL
                    {$$CONCEPT;Context;&Plane}:1 0))))
]

(* The Label is an optional part of the Window nestled either
above or below the Aperture inside the Border.
It is manifest by a special sort of Window which itself has no
Label but is drawn as Text on an inverted background.)

[roleset Label
  (diffs Realization@DISPLAY)
  (vr WINDOWLABEL)
  (derivation (MakeWindowLabel (Arguments $$CONCEPT))
              (Consequents InnerServant))]

(* The Location of the Window is the position of its lower
left corner on its Context.)

[roleset NIL
  (mods Location@DISPLAY)
  [derivation (MakeWindowLocation (Prerequisites Context Height Width)
              (Arguments (CAR $$CONCEPT;Context)
               (CAR $$CONCEPT;Height)
               (CAR $$CONCEPT;Width))]
]

(* The Width of the Window can be derived from any of several
other roles depending on what is already filled.
We will give as arguments all of the possibilities and let the
derivation function determine what to do.)

[roleset NIL
  (mods Width@DISPLAY)
  [derivation (MakeWindowWidth (Arguments (CAR

193 Source File: AIPSWINDOW
(* The Height of the Window can be derived from any of several other roles depending on what is already filled. We will give as arguments all of the possibilities and let the derivation function determine what to do. In addition, the Label can affect the Height of the Window if it is present.)

[roleset NIL
 (mods Height@DISPLAY)
 (derivation (MakeWindowHeight (Arguments (CAR
 [$$CONCEPT;Aperture;Height])
 (CAR
 [$$CONCEPT;Border;Height])
 (CAR
 [$$CONCEPT;Substrate;Height])
 (CAR
 [$$CONCEPT;Label;Height])))
]

[roleset NIL
 (mods Envelope@DISPLAY)]

[PROG (servant) ]
 (servant_[roleset Servant
 (number (0 NIL))
 (vr WINDOW)])

(* A Window may have other windows closely associated with it. The "Servant" relationship indicates other windows whose locations must be attended to when a window is moved.)

[roleset OuterServant
 (diffs (atomval servant))]  
[roleset InnerServant
 (diffs (atomval servant))
 (derivation (CopyWindowInnerServantFromLabel
 (Arguments [$$CONCEPT;Label]:1))))]

(* The role Servant is differentiated into InnerServant and OuterServant because the rules for relocating these two cases are likely to differ. Note that it is necessary to use a circumlocution in order to set up the differentiation properly (because of name inheritance.))
[roleset Master
    (vr WINDOW)]
(preferredprototype `NONSCROLLWINDOW)
(itags (ToDraw TDWindow)
    (ToLocate TLWindow)
    (ToSize TSWindow))))}

(InitWindowConcepts
    [LAMBDA NIL
        (InitAPERTURE)
        (InitWINDOW)
        (Init:NONSCROLLWINDOW)
        (InitSCROLLWINDOW)
        (InitTTYWINDOW)
        (InitSLOWWINDOW)
        (InitFASTWINDOW])
    )

(RPAQQ AIPSWINDOWUTILITYFNS (OpenWindow))
(DEFINEQ

/OpenWindow
    [DLAMBDA ((windowType Concept (SATISFIES windowType df `WINDOW))
        (substrate IndividualConcept (SATISFIES substrate df `VIEWSURFACE)))
        (* Edited by J.Gibbons on 29-Jan-81.)
        (PROG (window)
            (window (SendMessage windowType 'ToMake substrate):1)
            (SendMessage window 'ToSize)
            (SendMessage window 'ToLocate)
            (KLAddInstanceRole substrate Window@VIEWSURFACE <window>)
            (RETURN window))])

(RPAQQ AIPSWINDOWTOLOCATEFNS (TLAperture TLWindow))
(DEFINEQ

(TLAperture
    [DLAMBDA ((aperture IndividualConcept (SATISFIES aperture df `APERTURE))

195 Source File: AIPSWINDOW
(location (LST OF FIXP)))

(* The location of the Aperture itself must be (0 0) since it is a DisplayComposite. Otherwise, the location would translate the location of its parts which would be disastrous for the DisplaySurface.)

(if location or "location_{aperture;Border;Location}:1
then (if location
    then (SetRoleValues aperture Location@APERTURE '((0 0)))
    else location_ [aperture;&Border;&Location]:1)
(SendMessage {aperture;&Border}:1 'ToLocate location)
(SendMessage (CAR {aperture;&DisplaySurface})
    'ToLocate <location:1+1 location:2+1>)
    location])

(TLWindow
(DLAMBDA ((window IndividualConcept (SATISFIES window df ~WINDOW))
    (location (LST OF FIXP)))

(* Edited by J.Gibbons on 29-Jan-81.)

(if location or "location_{window;Location}:1
then (if location
    then (SetRoleValues window Location@WINDOW "location")
    else location_ [window;&DisplaySurface]:1
     do (printout T T "Window xCoord? 
        (xCoord_(RATOM))
     repeatuntil (FIXP xCoord)
     finally (RETURN xCoord_ (xCoord+8)/16*16))
    (bind yCoord
     do (printout T T "Window yCoord? 
        (yCoord_(RATOM))
     repeatuntil (FIXP yCoord)
     finally (RETURN yCoord))
    (SetRoleValues window Location@WINDOW <location>)
    (SendMessage {window;&DisplaySurface}:1 'ToLocate location)
    (SendMessage (CAR {window;&Aperture})
        'ToLocate <location:1+15 location:2>)
    location])
)

(RPAQQ AIPSWINDOWTOMAKEFNS (TMAperture TMPastWindow TMNonScrollWindow
    TMSlowWindow TMTTYWindow))

(DEFINEQ

Source File: AIPSWINDOW

196
(TMAperture
   [LAMBDA (genericAperture displaySurface planeFlg)
      (* Edited by J.Gibbons on
       27-Jan-81.)
      (Make APERTURE DisplaySurface (if planeFlg
         then (TMDisplaySurface NIL
            displaySurface 0)
            :1
            else displaySurface)
       Border 'Fill)
   >])

(TMFastWindow
   [DLAMBDA ((genericFastWindow GenericConcept (SATISFIES genericFastWindow
      df 'FASTWINDOW))
      (substrate IndividualConcept (SATISFIES substrate df
         'DISPLAYSURFACE))
      (* Edited by J.Gibbons on
       2-Feb-81.)
      (Make FASTWINDOW Substrate substrate Context 'Fill DisplaySurface
       'Fill Border 'Fill Aperture 'Fill)
   >])

(TMNonScrollWindow
   [LAMBDA (genericNonScrollWindow substrate)
      (* Edited by J.Gibbons on
       27-Jan-81.)
      (Make NONSCROLLWINDOW Substrate substrate Context 'Fill DisplaySurface
       'Fill Border 'Fill Aperture 'Fill)
   >])

(TMSlowWindow
   [LAMBDA (genericSlowWindow substrate)
      (* Edited by J.Gibbons on
       18-Dec-80.)
      (Make SLOWWINDOW Substrate substrate Context 'Fill DisplaySurface
       'Fill Border 'Fill Aperture 'Fill)
   >])

Source File: AIPSWINDOW
(TMTTYWindow
[DLAMBDA (((genericTTYWindow GenericConcept (SATISFIES genericTTYWindow df 
'TTYWINDOW))
(context IndividualConcept (SATISFIES context df 
'DISPLAYSURFACE)))
(* Edited by J.Gibbons on 2-Jan-81.))

(* Derives a NonScrollWindow for displaying TTY output.
Yes, this is a misnomer in that scrolling will occur but will
be entirely under control of BMG. This may be just one
indication that the hierarchy of windows needs rethinking.)

<(Make TTYWINDOW Context context DisplaySurface 'Fill Border 'Fill
Aperture 'Fill Substrate 'Fill)
>])

(RPAQQ AIPSWINDOWTOSIZEFNS (TSAperture TSNonScrollWindow TWindow))

(DEFINEQ

(TSAperture
[DLAMBDA (((aperture IndividualConcept (SATISFIES aperture df `APERTURE))
(height (ONEOF NIL FIXP))
(width (ONEOF NIL FIXP)))
(* Edited by J.Gibbons on 28-Jan-81.)
(if height or width or ~(height [aperture;Height]:1
and width [aperture;Width]:1)
then (if height
then (SetRoleValues aperture Height@APERTURE <height>)
else height [aperture;Height]:1)
(if width
then (SetRoleValues aperture Width@APERTURE <width>)
else width [aperture;Width]:1)
(SendMessage aperture;&Border]:1 'ToSize height width)
(SendMessage aperture;&DisplaySurface]:1 'ToSize height-2
width-2))

<TSNonScrollWindow
[DLAMBDA (((nonScrollWindow IndividualConcept (SATISFIES nonScrollWindow

Source File: AIPSWINDOW
(* Sets the Height and Width if the argument is supplied. Otherwise, they are derived if not already set. Setting of the values causes ToSize messages to be sent to the nonScrollWindow's parts. In any case, the values are returned. *)

(PROG ((substrate {nonScrollWindow;&Substrate}:l))
  (if height or width or -(height {nonScrollWindow;Height}:l and width_{nonScrollWindow;Width}:l)
   then (if height
         else height_2+[substrate;&Height]:l)
   (if width
         else width_32+[substrate;&Width]:l)
   (SetRoleValues nonScrollWindow Height@NONSCROLLWINDOW <height>)
   (SetRoleValues nonScrollWindow Width@NONSCROLLWINDOW <width>)
   (SendMessage {nonScrollWindow;&Aperture}:l 'ToSize height width-30)
   (SendMessage {nonScrollWindow;&DisplaySurface}:l 'ToSize height width))
  (RETURN <height width>))

(* Sets the Height and Width if the argument is supplied. Otherwise, they are derived if not already set. Setting of the values causes ToSize messages to be sent to the window's parts. In any case, the values are returned. *)

(TSWindow
 [DLAMBDA ((window IndividualConcept (SATISFIES window df `WINDOW))
   (height (ONEOF NIL FIXP))
   (width (ONEOF NIL FIXP)))])

(height_(RATOM)) repeatuntil (FIXP height))

(if width
   else (do (printout T T "Window width? ")
            (width_(RATOM)) repeatuntil (FIXP width)
            finally width (width+8)/16*16))
(SetRoleValues window Height@WINDOW <height>)
(SetRoleValues window Width@WINDOW <width>)
(SendMessage {window;&Aperture|:1 'ToSize height width-30)
(SendTime message {window;&DisplaySurface|:1 'ToSize height width})

(DECLAIM: DOEVAL@COMPILE DONTCOPY)

(ADDTOPAR GLOBALVARS AIPSUSERMODEL)
)

(ADDTOPAR CKLONEFILES AIPSWINDOW)
STOP