8 VIEWING

8.1. Introduction

This section describes the use of the „Viewing“-submenu. This is a group of commands, which are used to modify different parameters of the workcell views. (viewing position, view direction, zooming factor, size and position of the viewports on the graphics screen etc.).

8.2. The Viewing Panel

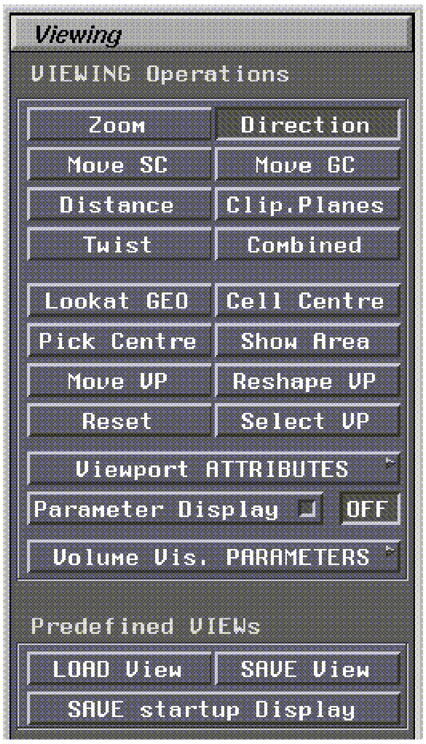


Figure 12: The Viewing Panel

- **Zoom**

  This function is used for „zooming“, i.e. the viewing angle (the opening angle of the virtual lens) is changed.

  - **LM** Change the viewing angle (zoom) for the main view
  - **MM** Change the viewing angle (zoom) for the camera overlay view

- **Direction**

  Allows to rotate the viewing position around the „point-of-interest“ (lookpoint). The line of sight will remain directed on the lookpoint. Additionally this valuator function allows to manipulate the field of view (zoom).

  - **LM** Change the azimuth angle

  - **MM** Change the elevation angle
  - **BM** Change viewing angle of the main viewport (zoom)

- **Move SC**

  This valuator function allows a parallel shift of the viewing position (viewpoint) and the point-of-interest (lookpoint). The viewing direction (azimuth and elevation angles) remains constant. The parallel translation of „viewpoint“ and „point-of-interest“ is carried out in the screen coordinate system (SC: screen coordinates).

  - **LM** Translate Right-Left
  - **MM** Translate Up-Down
  - **BM** Translate In-Out, i.e. along the line-of-sight

- **Move GC**

  This valuator function allows a parallel shift of the viewing position (viewpoint) and the point-of-interest (lookpoint). The viewing direction (azimuth and elevation angles) remains constant. The parallel translation of „viewpoint“ and „point-of-interest“ is carried out in the global world coordinate system (GC: global coordinates).

  - **LM** Translate in X-direction
  - **MM** Translate in Y-direction
  - **BM** Translate in Z-direction

- **Distance**

  This valuator function is used to change the distance between the the viewing position (viewpoint) and the point-of-interest (lookpoint). The viewing position is moved along the line-of-sight, i.e. the lookpoint position is not changed.

  - **LM** Change distance
  - **MM** No function
  - **BM** No function

- **Clip. Planes**

  Allows to move the „front“- and „back“ clipping planes, respectively. In STEREO-mode, this command allows additionally to change the stereo base (eye distance).

  - **LM** Move the front-clipping-plane
  - **MM** Move the back-clipping-plane
  - **BM** Change STEREO eye distance (in stereo mode only)

- **Twist**

This function rotates the view around the line-of-sight (Twist).

  - **LM** Change azimuth angle for side views
  - **MM** no function
  - **BM** no function

- **Combined:**

  This mode allows combined control of viewing position, direction and field-of-view (zoom)

  - **LM** Changes viewing position in screen coordinates in x- and y-direction
  - **MM** Changes azimuth and elevation angles of the viewing direction
  - **BM** Changes the field-of-view angle (zooming)

- **Lookat GEO**

  This PICK command allows fast centering of geometry parts. After triggering this command, a geometry element (GEO) is identified (picked) using LM. The „lookpoint“ of the new view is placed into the midpoint of the bounding box covering the identified GEO.

  - **LM** PICK the GEO element
  - **MM** Picking is stopped (no change of viewing parameters)
  - **BM** BM not used

- **Cell Centre**

  The lookpoint (point-of-interest) is moved into the arithmetic model centre. The centre is calculated as the midpoint of the bounding box covering all geometry parts in the model.

- **Pick Center**

  This PICK-function is used for fast redefinition of the point-of-interest coordinates. LM is used to PICK the new view direction in the workcell display. The viewing transformation matrix is recalculated as follows:

  - the viewing location is not changed (for perspective views)
  - for orthonormal projections, view- and lookpoint are translated
  - a new line-of-sight is computed, using the PICK position as direction
  - the PICK location becomes the new centre of the viewport (in model space)
  - the distance between viewpoint and previous lookpoint is not changed

- **Show Area**

  Allows a fast redefinition of the new 3D-window. This function is a combination of „Zoom“ and „Center View“. After pressing LM, the current cursor position is used as one corner of the new viewing frustum. Now you can drag a rubberband rectangle (keep LM pressed down) to define the opposite corner of the view area. The aspect ratio of the world window is not changed and the original viewport coordinates (screen space) are used.

- **Move VP**

This function is used to move overlay **v**iew**p**orts (PAN) for multiviewport display modes. The dimension of the viewport is not changed. Press LM over the viewport you want to move. Keep LM pressed and move the rubberband rectangle to the new location.

- **LM** Used to identify and move the overlay viewport
- **MM** No function
- **BM** No function

- **Reshape Viewport**

  This function is used to relocate and resize overlay viewports for multiviewport layout configurations. Use **SELECT VP** to identify the viewport and release it. Then press **Reshape VP**. Use LM to define one corner of the new viewport location. Keep LM pressed and draw the rubberband rectangle to define the opposite corner.

- **Reset**

  Resets for all defined views the viewing parameters to the values as predefined in the layout-file (‚.lay‘). This is, the viewing parameters are set to the initial state as when KISMET is started for a particular simulation model.

- **Select VP**

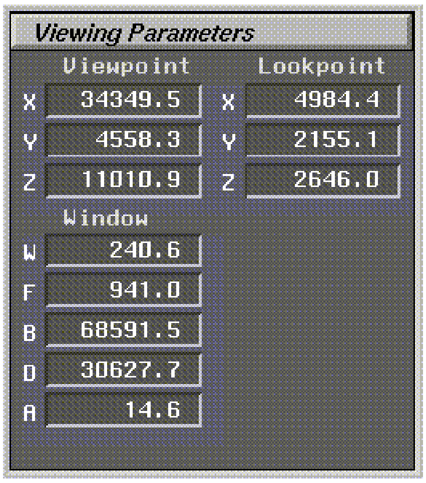
  Defines the input focus for viewing commands in multiviewport display configurations. Input focus is toggled from the big main viewport (e.g. top view) to the overlay viewport (side view) and vice versa.

- **Viewport Attributes**

  Leads to the Viewport Attributes Panel. See chapter 8.2.1 Viewport Attributes for details.

- **Parameter Display ON/OFF**

  Display of Viewing Parameter.



Viewing Parameters

Viewpoint		Lookpoint	
X	34349.5	X	4984.4
Y	4558.3	Y	2155.1
Z	11010.9	Z	2646.0

Window

W	240.6
F	941.0
B	68591.5
D	30627.7
A	14.6

Figure 13: Viewing Parameters

- **Vol. Vis. Parameters**

  Leads to the Volume Visualisation Panel. See chapter 8.2.2 Volume Visualisation Parameters for details.

- **LOAD View**

  This function is used to load a workcell display from a binary ’view’-file. The filename is picked from a file-select-panel. To create a ’view’-file use the **SAVE View** command.

  The function is used for fast reseting of all workcell views. A similiar command is available for the SCRIPT Interface ("LOAD_VIEWS").

- **SAVE View:**

  The function is used to save the current workcell display (all views) in a binary ’view’-file. The pathname for the ’.view’-file is *"$kis_home/scripts/<view_filename>"*. The default extension is ’.vws’. The file can be used for fast reset of a workcell view, using the **LOAD View** command. ’View’-files are useful for animation because you can load them from within a SCRIPT.

- **Save startup Display**

  This function writes the current workcell display parameters to a LAYOUT-file in ASCII-format. This file is used by KISMET every time during simulation startup. The LAYOUT file (default extension ’.lay’) defines the basic display layout, i.e. the size and location of all viewports and their default viewing parameters.

8.2.1 Viewport Attributes

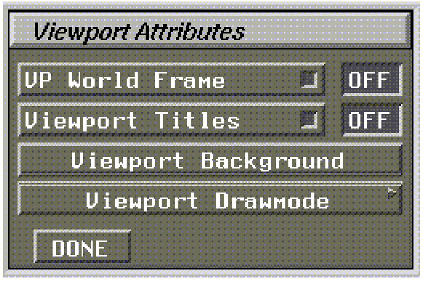


Figure 14: Viewport Attributes

- **VP World Frame ON/OFF**

  World Frame will be displayed or not in the selected viewport.

- **Viewport Titles ON/OFF**

  Selected viewport will have a title or not.

- **Viewport Background**

  Selection of background colour of the chosen viewport

- **Viewport Drawmode**

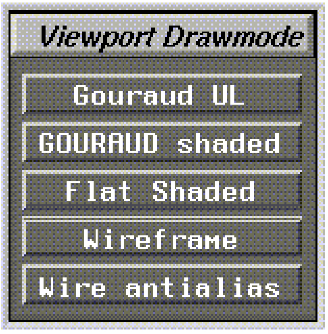


Figure 15: Viewport Drawmode

Selection of the desired drawmode of the selected viewport. See 9.2.1 Draw Parameters for a description of the available drawmodes.

8.2.2 Volume Visualisation Parameters

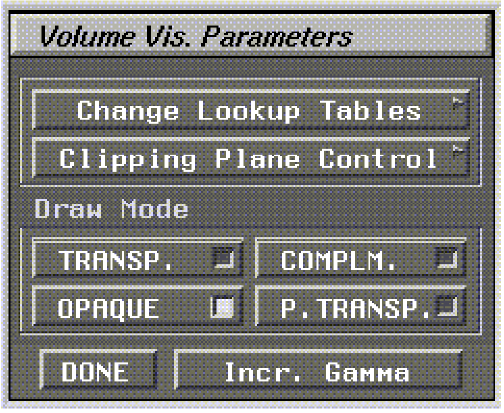


Figure 16: Volume Visualisation Parameters

These buttons are only functional in case a KISMET compatible volume file ('*.mpo') is loaded.

8.2.2.1 Change Lookup Tables

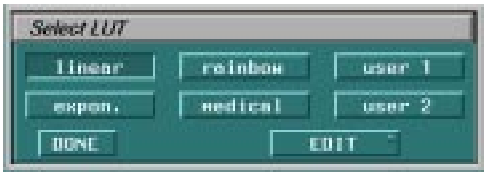


Figure 17: The "Change Lookup Tables Menu"

This set of buttons is used to select the desired lookup table (LUT, also called 'transfer functions') for the volume visualization. Lookup tables are used to translate original incoming datavalues into other, specially chosen outputvalues. (e.g. densitiy information to transparency).

- **linear**

  Selects the linear LUT. Incoming values are mapped without change to outgoing values.

- **expon.**

Selects the exponential LUT. Incoming values are translated according to an exponential curve.

- **rainbow**

  Selects the artificially colored LUT. Incoming greyvalues are translated into colored outgoing values. The characteristics of the LUT can be changed by editing the 'rainbow.klt' file. (see"USER DATA SPECIFICATION" for detailsl)

- **medical**

  Selects the medical LUT. Especially precalculated for the visualization of medical MRI datasets for diagnostic purposes.

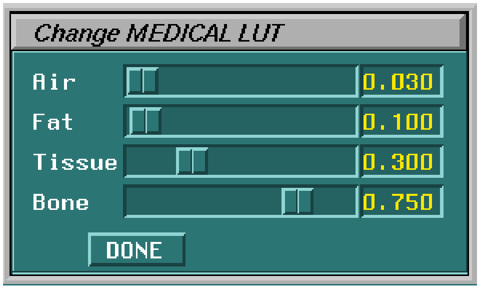
- **user1/user2**

  Selection of two user defineable LUTs. The filenames must be 'user1.klt' and 'user2.klt'. For the creation of user-LUTs and the manipulation of predefined LUTs please check the technical manual.

- **EDIT**

  This button leads to a separate editing window for the LUT parameters. Which editing window comes up depends on the selected LUT:

  - **Selected LUT: linear / expon. / rainbow / user 1 / user 2**



Change MEDICAL LUT

Air 0.030

Fat 0.100

Tissue 0.300

Bone 0.750

Figure 18: The "Change Medical Lookup-Table " Panel (1)

Alpha

This slider changes the transparency factor of the visualization.

Gain

This slider influences the overall intensity of the whole dataset.

Level

This slider selects a certain point on the translation curve as the upper end of the output-value window. (See diagram)

Window

This slider selects the opening width of the output-value window. (See Figure 19: Width of Medical LUT output-value window)

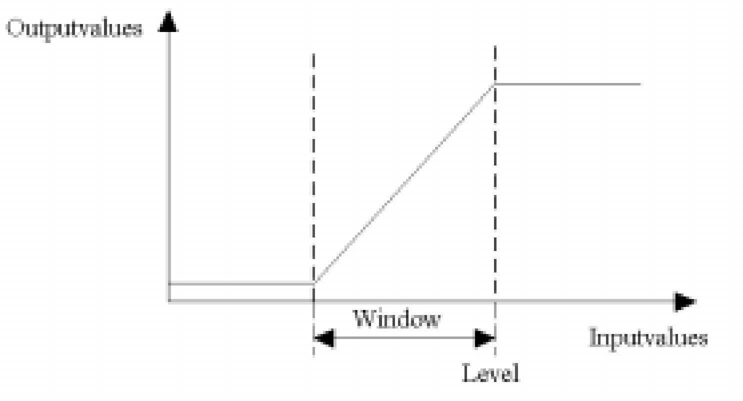
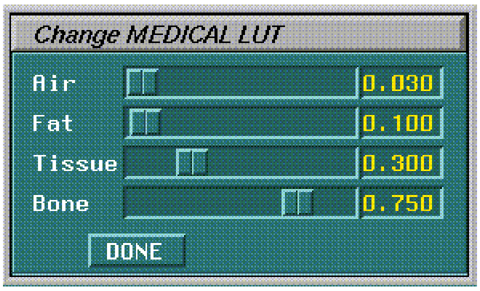


Figure 19: Width of Medical LUT output-value window

The original outgoing values from the selected LUT (linear, exponential...) are multiplied by the outputvalues from this window-leveling function.

- **Selected LUT: medical**



Change MEDICAL LUT

Air 0.030

Fat 0.100

Tissue 0.300

Bone 0.750

Figure 20: The "Change Medical Lookup-Table " Panel (2)

The medical LUT has a special set of parameter values, which is in the current version optimized for the representation of MRI (Magnetic Resonance Imaging) data.

Air

Changes the visualization intensity of the air, surrounding the object.

Fat

Changes the visualization intensity of fatty tissue.

Tissue

Changes the visualization intensity of soft tissue (muscles, skin).

Bone

Changes the visualization intensity of bone.

8.2.2.2 *Clipping Plane Control*

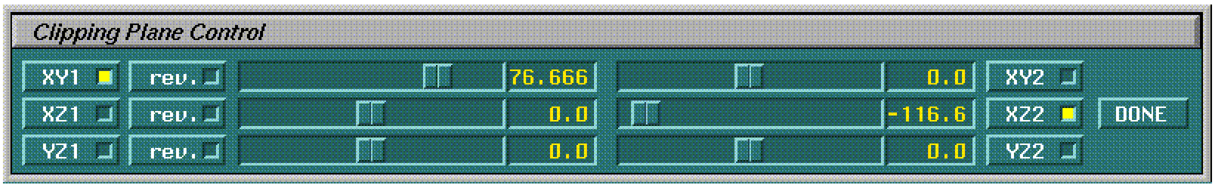


Figure 21: The "Clipping Plane Control"-Panel

This is the control center for the 6 possible orthogonal hardware clipping planes. The first set of three planes can be switched on and off by pressing the 'XY1', 'XZ1', 'YZ1'. The 'rev.' button may be used to invert the clipping plane's normal vectors. Every plane has a slider to move the plane through the volume. The distance between the plane and the centerpoint of the volume is shown in milimeters.

Additionally it is possible to activate the second set of three clipping planes by pressing the buttons 'XY2', 'XZ2', 'YZ2'. Reverse configuration of these three planes makes no sense and is therefore not implemented.

8.2.2.3 *Draw Mode*

- **TRANSP.**

  The volume is shown transparent. The grade of transparency can be influenced by the sliders in the 'Select LUT - Change Medical LUT' box.

- **OPAQUE**

  The volume is drawn opaque. The 'Alpha'-slider in the 'Change MEDICAL LUT'-box is not functional.

- **COMPLM.**

  The part of the volume which has been clipped away by applying the clipping planes is shown. So it is possible to cut out rather complicated geometries.

- **P.TRANSP.**

  The part of the volume which has been cut away by applying clipping planes is drawn transparent, whereas the rest of the volume is shown opaque.

- **DONE**

  leave the dialogue

- **Incr. Gamma**

  This button allows to correct the gamma value slightly if the volume drawn is too dark to be seen comfortably. When the button is pressed, the text changes to 'Reset Gamma' and the former gamma state may be restored after pressing it once again.

9 DISPLAY

9.1. Display: Introduction

This submenu is used for selection of different screen layout options, to control some display options.The main display modes do not affect the CAMERA-view overlay (K-Key).

The default display layout is set at model startup time by reading the layout-file ('.lay') of the application.

9.2. The Display Panel

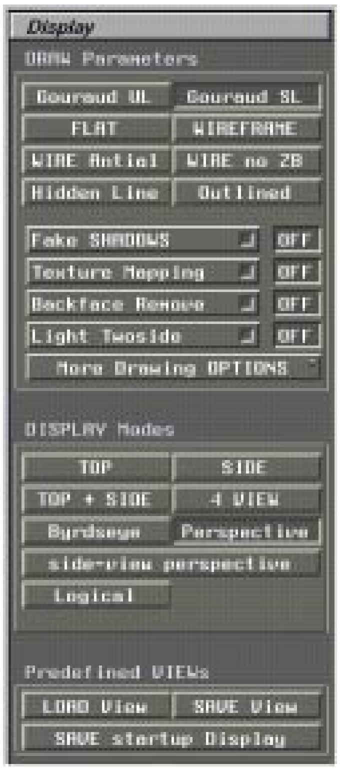


Figure 22: The Display Panel

9.2.1 Draw Parameters

The Draw Parameters are used to select between different rendering qualities in KISMET. You can select various types of shaded rendering, wireframe rendering and other image quality features.

Note: The quality and performance of the different rendering modes hardly depends on the graphics hardware of your workstation.

- **Gouraud UL**

  This render mode (**GOURAUD**, **U**ser **L**ights) is set to GOURAUD shaded rendering with user-defined light model. The light model is specified in the lighting-file (‚.lgt') of the simulation model database. Up to eight lightsources can be specified. The FastKey F11 is used to switch local viewer mode ON. The FastKey F12 puts KISMET back to infinite viewer lightsource model. The default is infinite viewer.

  Note: Refer to your workstation documentation to see if the graphics hardware is capable of doing GOURAUD-shading.

- **Gouraud SL**

  This render mode (**Gouraud**, **S**ystem **L**ight) selects GOURAUD-shaded rendering with the graphics system lightsource, i.e. the parallel source is always located in the viewpoint, directing along the line of sight (like a miner’s lamp).

- **FLAT**

  The drawmode is set to FLAT-shaded rendering. All pixels in a surface polygon are filled with the same, calculated color. This mode is using one lightsource in the viewpoint.

- **WIREFRAME**

  Select the wireframe-drawmode with Z-buffering. No shading or lighting is performed. Although, the GEO-elements are displayed in the correct depth order.

- **Wire Antial**

  The drawmode is switched to **wireframe-antial**iased-mode. The lines are displayed antialiased (if your graphics hardware allows this option). This gives a smoother appearance of the polylines. No shading or lighting is done.

- **Wire no ZB**

  Set drawmode to **wire**frame-drawmode without **Z-b**uffering. No shading or lighting is performed. The GEO-elements do not appear in the correct depth order

- **Hidden Line**

  The scenario is drawn in a Pseudo-Hidden-Line-mode. This is, all facets are outlined with BLACK-colour. The facet insides are filled with the background colour. The result appears on most hardcopy devices like a hidden-line wireframe drawing.

- **Outlined**

  Selects flat-shaded rendering outlined polygon edges. This mode may be useful for enhancing the contrast for hardcopy output.

- **Fake SHADOWS ON/OFF**

  Enables/disables the display of fake-shadows. Shadows of all model parts are drawn in the plane (Y=0). This method is based on a projection technique. There are no shadows generated between the objects, but it is capable of near-realtime performance.

  Note: For full shadows you may use the RAYCASTING-mode (F5-Key). Reflections between the objects are calculated additionally in RAYTRACING-mode (F6-Key). Be careful if you try this, it may take hours to compute just on image !

- **Texture Mapping ON/OFF**

  Enables/disables the display of surface-TEXTURES (hardware texture mapping). KISMET allows mapping of SGI image files (in ‚rle‘ format) on geometric surfaces, this is it works for EXTENDED_POLYHEDRON geometry parts only.

  Texture mapping in KISMET works as follows:
  The texture file(s) is (are) defined in the ’.mat’ file of the application model. The material index of the GEO-part refers to those materials which have defined texture files. A texture list is defined in the EXTENDED_POLYHEDRON ‚.mpo‘ file. The texture list is used for

the mapping from 3D-object space to 2D texture space.

On some workstations texture mapping is done in realtime or near-realtime (hardware supported). On other workstations this function is emulated in software (with the resulting poor performance) or it is not working at all. Please refer to your workstation documentation if texture mapping is available on your model. The texture mapping function in KISMET is by default OFF.

- **Backface Remove On/OFF**

  Toggles function for hardware "BACKFACE-REMOVAL" in the graphics subsystem. For some workstations, enabling backface removal can improve the graphics performance of KISMET to some extent. This is, using this option you may be able to achieve higher display rates. The default value is OFF (disabled).

- **Light Twoside On/Off**

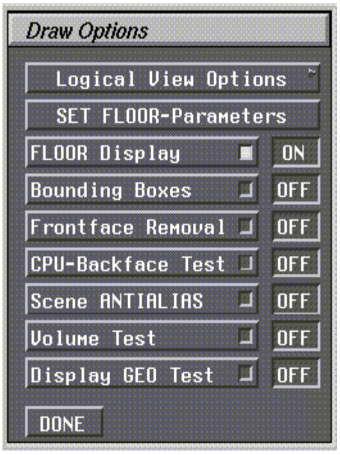
  Enables/Disables the lighting of backfacing polygons. With twoside lighting mode active (On), the rendering performance is substantially reduced. The default value is OFF (disabled).

- **More Drawing Options**

  Another submenu panel for addional, less frequently used rendering options is called. See chapter 9.2.1.1 More Drawing Options for available options.

9.2.1.1 More Drawing Options

The Draw Options Panel:



Draw Options

Logical View Options

SET FLOOR-Parameters

FLOOR Display ■ ON

Bounding Boxes □ OFF

Frontface Removal □ OFF

CPU-Backface Test □ OFF

Scene ANTIALIAS □ OFF

Volume Test □ OFF

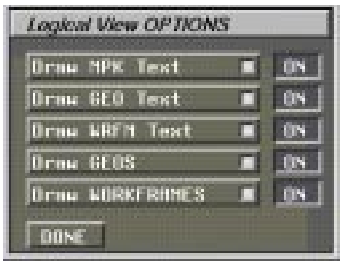
Display GEO Test □ OFF

DONE

Figure 23: The Draw Options Panel

- **Logical View Options:**

  Leads to the Logical View Options Panel. The menu switches define various rendering options in logical viewing mode. For a general description of the locical view, please refer to chapter 9.2.3.1 Locical View. The various display options controlled by the menu are :



Logical View OPTIONS

Draw MPK Text ■ ON

Draw GEO Text ■ ON

Draw WRFM Text ■ ON

Draw GEOS ■ ON

Draw WORKFRAMES ■ ON

DONE

Figure 24: The Logical View OPTIONS Panel

•Draw MPK Text

Toggles display of frame names

•Draw GEO Text

Toggles display of geometry (.mpo file) names

•Draw WFRM Text

Toggles display of WORKFRAME names

•Draw GEOS

Toggles display of geometry symbols (includes the names)

- **Draw WORKFRAMES**

  Toggles display of Workframe symbols (includes the name)

- **Set Floor Parameters**

  This function leads to a set of panels which are used to define or change the drawing parameters of the FLOOR object. To save the attributes in your model database, you have to write the ABS-root file to the file-system. In the definition panels, you can define the placement, size and colour scheme of the FLOOR-geometry. The FLOOR can be drawn uniform or as a two colour rectangular grid (chessboard like).

- **Floor Display ON/OFF**

  Enable/disable the display of the FLOOR-object. The FLOOR is an automatically generated object. It is a special kind of WFRM-data. The appearance can be defined in the root ’.mpc’-file

- **Bounding Boxes ON/OFF**

  This command is used to toggle the display function for the overlay of BOUNDING-BOXES. They are used within the collision test algorithm for the first two stages of the test. Each GEO part is surrounded by a box (drawn in CYAN). Alle parts of a FRAME are enclosed in another box (BLUE). The higher-level FRAMEs (father frames) are again surrounded by a box and so on. See chapter 18 Function selection panel for details concerning the collision test option.

- **Frontface Removal ON/OFF**

Toggles the display mode for graphics hardware "FRONTFACE REMOVAL", i.e. all polygons facing to the viewer are not rendered. The default value is "Off".

- **CPU-Backface Test ON/OFF**

  Toggles function for software"BACKFACE-REMOVAL". For some workstations, enabling software backface removal can improve the graphics performance of KISMET to some extent. This is, using this option you may be able to achieve higher display rates. The default value is OFF (disabled).

- **Scene ANTIALIAS**

  Triggers antialised drawing for the next rendering frame (one-shot). Due to the chosen method in KISMET (the scene is rendered in the invisible backbuffer several times with minimal random shifts in the viewing position, the subframes are accumulated to the final image), this function may consume some rendering time.

- **Volume Test**

  Toggles testing of each polygonal facet of all geometrical entity before drawing according to the visible subvolume of the scene to improve drawing performance. The default value is "Off".

- **Display GEO Test**

  Toggles Volume Testing of geometrical parts for improving the rendering performance. In active mode, the bounding boxes for every geometric entity in the scene are tested for visibility before rendering all polygonal faces of the geometric part. The default value is "Off".

9.2.2 Display Modes

- **TOP**

  Selects a single „top“-view display layout. The line-of-sight is parallel to the negative Y-axis of the world coordinate system. Orthonormal projection is used.

- **SIDE**

  Enables a screen layout with one orthonormal „side“-view. The line-of-sight is perpendicular to the XZ-plane of the world system. It can be rotated around the Y-axis.

- **TOP + SIDE**

  The KISMET screen layout is set for display of a „top“-view with a „side“-view overlay window. Both views use orthonormal projection. The line-of-sight lies along the negative Y-axis of the model world system in the top-view. It is perpendicular to the XZ-plane of the world system in the side-view. The „Viewing“-functions are limited in both windows.

- **4 VIEW**

  The KISMET screen layout is spltting the viewport into 4 different views. You will get TOP-, SIDE- and Perspective-View together with a Front-View.

- **Birdseye view**

  Enables a workcell display with perspective projection. The „Viewing“-commands simulate a fly-through-the-scene appearance. „Birdseye view“ allows control of all viewing attributes. The „Viewing“-command „Change View-Direction“ rotates in this configuration the lookpoint, whereas the viewing position remains fixed.

- **Perspective**

  Enables a workcell display with perspective projection. This mode allows control of all viewing parameters with the „Viewing“-functions. The „Viewing“-command „Change View-Direction“ rotates in this configuration the viewpoint, whereas the point-of-interest remains fixed in space. This differs from the behaviour in „Birdseye“-mode.

- **side-view perspective**

  Selects a perspective „side“-view projection. The other attributes are similar to the „single side-view“.

- **Logical**

  Selects the Locical Viewing Mode of the scenes datastructure. In this display mode, symbols are use to display ABSTRACT-files, MPK-frames, geometrical parts and workframes. This mode is sometimes better suited for selecting (picking) the models entities than geometrical rendering. It gives a good overview of the hierarchical structure of the KISMET dataset. Please refer to chapter 9.2.3.1 Locical View for a more detailed description of the symbols.

9.2.3 Predefined VIEWs

- **LOAD VIEW**

  This function is used to load a workcell display (all views) from a binary VIEW-file. The filename is picked from a file-select panel. To create a VIEW file use the **SAVE VIEW** button.

  The function is used for fast reseting of all workcell views. A similar command is availbale for the script interface ("LOAD_VIEWS").

- **SAVE VIEW**

  This function is used to save the current workcell display (all views) in a binary VIEW-file. The pathname of the view file is "$kis_home/scripts/<view_filename>". The default extension is ’.vws’. The view file can be used for a fast reset of all workcell views, using the **LOAD VIEW** command. View files are useful for animation, because you can load them from within a SCRIPT.

- **SAVE startup Display**

  This function writes the current workcell parameters to a LAYOUT-file in ASCII-format. This file is used by KISMET every time during simulation startup. The LAYOUT-file (default extension ’.lay’) defines the basic display layout, i.e. the size and location of all viewports and their default viewing parameters.

- **Logical View**

  Activates logical viewing mode, i.e. the model datastructure is rendered using symbols for frames, geometry, ABSTRACT files and workframes. See 9.2.3.1 Locical View for details.

9.2.3.1 Locical View

Purpose

Most of KISMET's viewing modes like

- TOP
- SIDE
- TOP + SIDE
- 4 VIEW
- Byrdseye
- Perspective
- Side-view perspective

are designed to show the geometric structure of models in form of different 3D-projections.

The "Logical View" viewing mode on the other hand is a means to visualize the logical file-structure of an actually loaded model tree.It should help the user to create new as well as to change existing models in a more convenient way.The most important purpose is to locate branching points in the model tree hirarchy and to locate and identify kinematic frames and the attached workframes and geometry objects.

Screen Layout

The following symbols are used to represent the logical structure of the model tree:

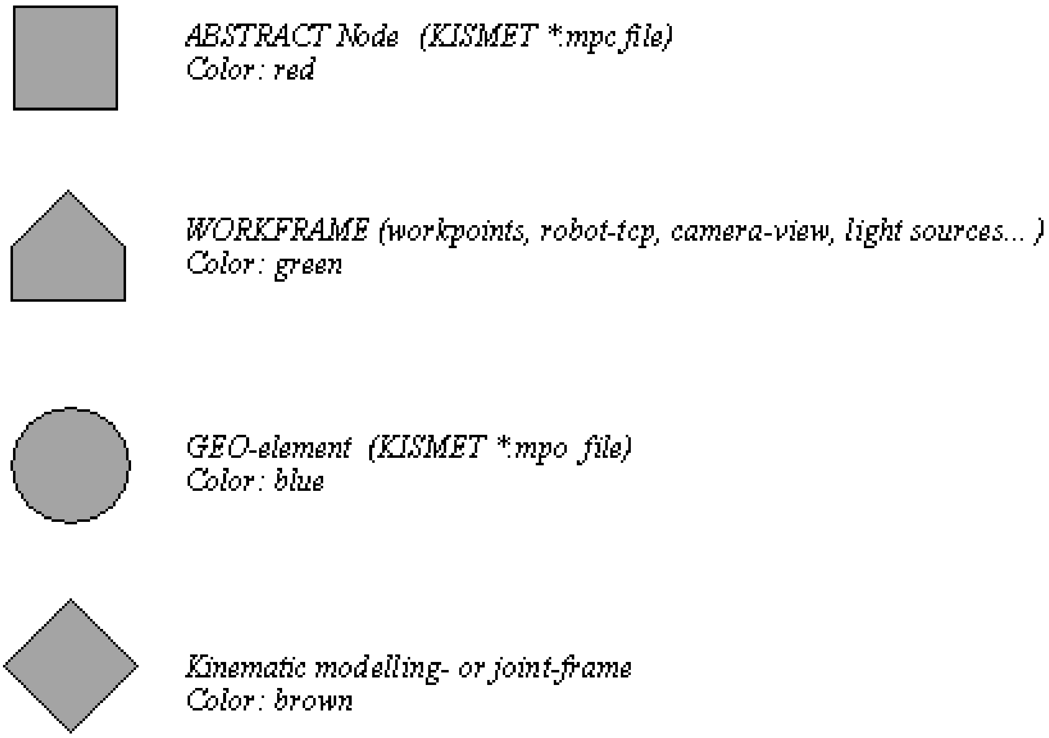

ABSTRACT Node (KISMET *.mpc file)
Color: red
WORKFRAME (workpoints, robot-tcp, camera-view, light sources...)
Color: green
GEO-element (KISMET *.mpo file)
Color: blue
Kinematic modelling- or joint-frame
Color: brown

Figure 25: Logical View Symbols

Example of a small model-tree

The following example shows a small model tree with two hierarchical steps.:

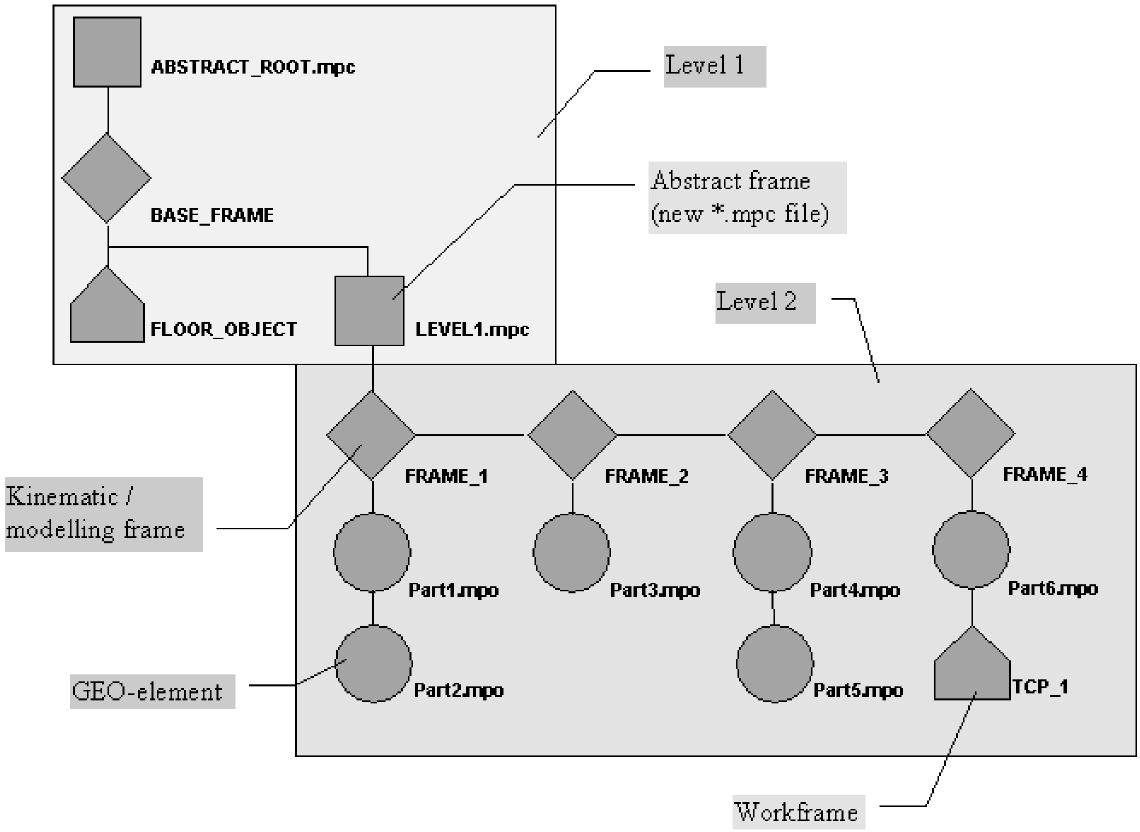


Figure 26: Display example for "Locical view"

Usage

- **Selecting "Logical View" mode**

  Press the "Logical" button in the "DISPLAY Modes" panel under the "DISPLAY" menu button.

- **Display features**

  There are the following options to influence the appearance of the Logical View graph:

  - switch off/on kinematic frame names
  - switch off/on drawing of GEO symbols
  - switch off/on GEO names
  - switch off/on drawing of WORKFRAME symbols
  - switch off/on WORKFRAME names

  To achieve a better view of the fundamental file-structure, disable the GEO- and WORKFRAME symbols.

  To find the appropriate control buttons follow this user interface hierarchy:

  - press the "DISPLAY" menu button
  - press the "More Drawing OPTIONS" button in the "Draw Parameters" panel
  - press the "Logical View Options" button in the "Draw Options" popup window

  - chose the desired buttons out of the "Logical View OPTIONS" popup window

  close the popups by pressing the "DONE" buttons.

- **Viewing operations**

  The functionalities of all the viewing operations while "Logical View" is enabled, is identical to those of most of the other KISMET viewing modes. The viewing operations buttons are located in the "VIEWING Operations" panel under the "VIEWING" menu button.

  To adapt the "Logical View" to your needs, use the following viewing operations .

  - **Zoom**

    Press the left mouse button and shift the mouse left /right to zoom in/out.
    Move SC (move screen coordinates)
    Press left mouse button and shift mouse left/right to move the graph left/right.
    Press middle mouse button and shift mouse left/right to move the graph up/down.
    Press left and right mouse button and shift mouse left/right to zoom in/out.

  - **Show Area**

    To get a good survey of the graph's logical structure, you can zoom out until the whole graph is to be seen. After that you can magnify the region of interest by pressing the "Show Area" button and select the region with a rubberband.

  - **Reset**

    Reset the view to the initial state.

  - **Modelling operations on the data tree**

    The "picking" functionality can also be used in the Logical View, so that it is possible to perform all the picking-operations which are known from the other viewing-modes.

    - **operations on GEO-elements (KISMET *.mpo files)**
      All the operations offered by the buttons in the "Edit Geometry (GEO)" panel (to find this press the "MODEL"-menu button ; choose the "Edit GEO"- button in the "Cell Modelling" panel) can be used to influence the GEO-elements in the model.
      You can for example delete, load, or mark any GEO-elements by picking the desired GEO-symbol in the Logical View display. After switching back to one of the projection views, the changes should be visible immediately.
      To find the file name related to a special GEO-element, pick the desired element in one of the projection view modes (e.g. "Perspective"), then switch to Logical View. The selected GEO-element will be marked there.

    - **operations on kinematic modelling FRAMES**
      All the operations offered by the buttons in the "Edit Parts (FRAME)" panel (to find this press "MODEL"-menu button ; choose the "Edit FRAME"- button in the "Cell Modelling" panel) can be used to influence the kinematic frames in the model.
      Just pick one of the brown frame symbols and perform the desired operations on it. If you decide for example to add a new abstract file (*.mpc) to one of the frames, the Logical View will be updated immediately.

    - **operations on WORKFRAMES**

All the operations offered by the buttons in the "Edit WFRM", "Edit WFRM-PATH" and "WFRM-File Operations" panels (to find this press "WFRM"-menu button) can be used to influence the workframes in the model.
Just pick one of the green workframe symbols and perform the desired operations on it.

- **DETAILING operations**
  All the operations offered by the buttons in the "Interactive Detailing" and "Edit Operations" panels (to find this press "DETAIL"-menu button) can be used to influence the complexity of the model.

ATTENTION !

To select one of the ABSTRACT-Frames (KISMET *.mpc files) please pick the kinematic modelling frame (brown rhombus) ABOVE the desired ABSTRACT-symbol (red square).

10 SIMULATE

10.1. Virtual IRDATA-Controller

KISMET provides a virtual robot controller for each robot (mechanism) defined in the model datafiles. The virtual controller stores the robot dependent parameters and variables. This is, for example current speed, accelleration for the TCP (linear movements) and joints (PTP move) TCP- and ZP-offsets, pointers to loaded robot programs etc.. Figure 27 shows the architecture of the virtual multi-tasking robot program controller as implemented in KISMET.

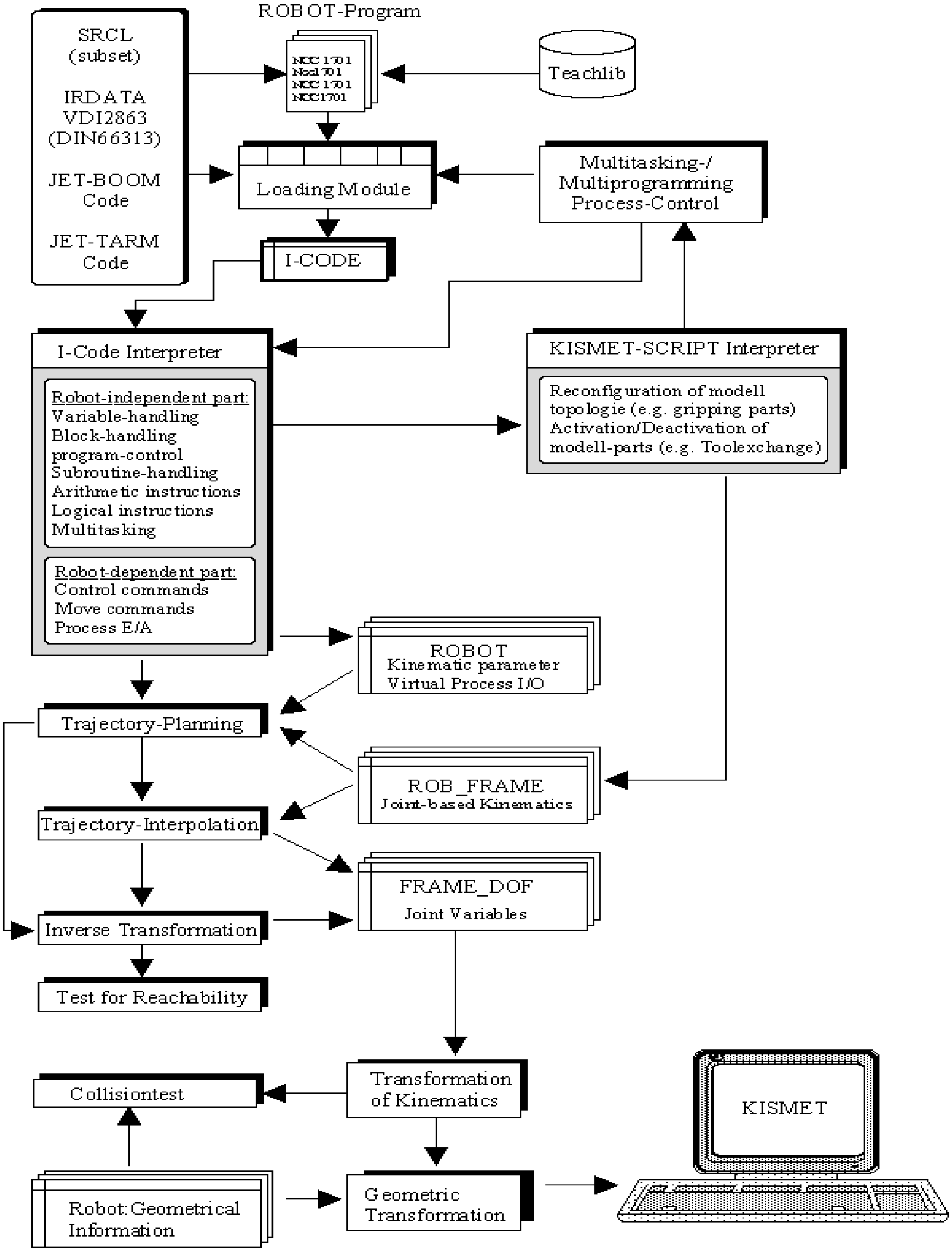


Figure 27: Architecture of the virtual robot programm controller

The program simulator has the following main features:

- The robot program (IRDATA-program) is loaded for the active ROBOT
- programs are read from the directory ’$kis_home/teachlib’
- for each ROBOT, multiple programs can be loaded (IRDATA)
- programs can be executed simultaneously, i.e. each robot in the workcell can execute a program
- in IRDATA, one program can control multiple robots (USEIR command in IRDATA)
- simultaneously executed programs can be synchronized using IRDATA semaphores (SIGNAL, WAIT, SEMINI commands)
- program execution can be performed in realtime mode or non-realtime mode
- realtime-mode means, program execution in KISMET takes the same time as on the real controller. The system clock (monitor intervalls) are used for this mode
- in non-realtime-mode, a timebase is defined, and program execution is performed at sampled timesteps. For every program- and display cycle, the virtual execution time is incremented by the user-defined time interval (timebase). This mode is more accurate as the realtime mode. The timebase can be defined in the range 1-5000 ms.
- KISMET can perform robot program execution either in continuous execution mode ("Run"), or in single program step mode ("Verify").
- The current excution time of the program is displayed in the output field "15" of the KISMET display (in non-realtime mode only). If multiple programs are executed, the execution time for the active ROBOT is shown. This output can be used for cycle or execution time measurements.
- The current step number (command number) of the active ROBOT is displayed in output field "7".

The simulation is based on a KISMET-internal datastructure (I-Code), which is interpreted by the virtual controller during robot-program execution. Read-modules build the interface between different robot languages or -codes and the I-Code.

The I-Code corresponds in its functionality to the IRDATA standard with some minor restrictions, but also some functional extensions. For example, it is possible to embed KISMET SCRIPT-commands into IRDATA comment lines. This method is used to include simulation specific commands into executable robot programs. The real robot controller interprets these lines as comments, whereas KISMET can filter useful commands out.

SCRIPT-commands are recognized by KISMET, when a comment line starts with a "%" character in the first position. Then the rest of the comment is interpreted as a SCRIPT command.

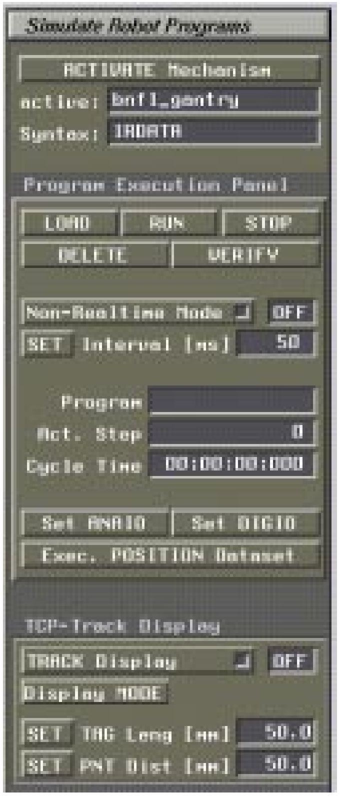
Example:

```
00011,21400,16,4,0,132,1,578.0,132,1,306.5,132,1,-10.8,
132,1,0.0,132,1,0.0,132,1,-180.0;
00012,5000,344,16,4,0,1,0,1,0;
00013,1000,8,‘Greife Schlagschrauber‘;
00014,1000,9,‘%CONNECT Traegersystem.WKZ_BAS.WRENCH
TO Traegersystem.J_2.HAND;
```

```
00015,21400,16,4,0,132,1,940.0,132,1,306.5,132,1,-10.8,
132,1,0.0,132,1,0.0,132,1,-180.0;
00016,5000,344,16,4,0,1,0,1,0;
```

The IRDATA code segment, as listed above, executes two linear motion commands to (commands 11, 12) and from (commands 15, 16) the tool pickup position of the end-effector named WRENCH. Commands 13 and 14 are comment lines. In the comment line 14, a SCRIPT-command is embedded for the kinematics module of KISMET. This SCRPT command in the example is used to connect the tool 'WRENCH' to the robot hand.

10.2. The Simulate Menu



active: bnfl_gantry
Syntax: IRDATA
Non-Realtime Mode OFF
SET Interval [ms] 50
Program
Act. Step 0
Cycle Time 00:00:00:000
TRACK Display OFF
SET TAG Leng [mm] 50.0
SET PNT Dist [mm] 50.0

Figure 28: The Simulate Menu

- **Activate Mechanism:**

  Selection of the active mechanism which can be a robot or any other kinematic structure, as well as the selection of the programming language to be used.

  The selcted mechanism is displayed in the *active*-field, while the selected language is visible in the *Syntax*-field.

10.2.1 The Program Execution Panel

This panel is used for simulated robot program execution. It includes the commands for robot program load, start, stop, verify and delete.

- **LOAD**

  A robot program is loaded for subsequent off-line simulation. During reading, the program is checked for syntactic correctness and an internal data structure (I-code) is created within KISMET. The I-code is robot language independent. If any error is encountered by KIS-

MET during program read, the ’vi‘-editor is started. The cursor shows the program line where the error was detected. You can leave ’vi’ using the ’:q’ command.

For each robot program loaded, a menu entry is created in the robot program submenu. The menu is involved during program selections ("Run", "Stop", "Delete", "Verify").

Note: The program to be loaded is assigned to the active robot, i.e. it is read into its virtual controller.

- **RUN:**

  Starts a program for off-line simulation. The program to be started is selected from the "program"-menu, which is created during program "Load".

- **Stop:**

  Stops a off-line simulated program during execution. It can be used after "Run" or "Verify". After the program is stopped, the program counter is set to the first command. This is, after a stop the program can be restarted at the beginning.

  The "Stop" selection is made from the "program"-menu. Each running program must be stopped separately. This means, if one program is stopped, any other programs will continue execution.

- **Delete:**

  A loaded program is deleted from the virtual controllers memory. The memory allocated during program "Load" and occupied by the I-code is deallocated.

  Note: Using "Delete Teachfile", the robot program is not erased from the UNIX-filesystem. In case you want to delete the program from the filesystem, you should use the UNIX-command *rm <filename>*.

- **Verify:**

  Starts a robot program, or continues a running program in single step mode. A special menu will appear in "Verify"-mode after each program step. See chapter 10.2.2 Verify-Submenu.

- **Non-Realtime Mode**

  Toggle function for robot program off-line simulation in realtime or non-realtime mode. The default is realtime mode.

  It is an advantage of the realtime that the response time nearly corresponds to the reality. This mode is useful when KISMET is used for operator training (Remote Handling -surveillance). On the other hand, important details of robot movements often cannot be observed and possible collisions are not detected during off-line simulation, in particular when the image rate is low due to large model size. In the latter case, non-realtime off-line simulation is to prefer. Fortunately, KISMET gives both options.

- **SET:**

  Is used for specification of the sample rate for non-realtime program simulation. The valid range for the timebase is in the wide band between 1 ms up to 5 seconds.

- **Set ANAIO:**

  Is used to set analog I/O-channels to a value. The virtual robot controller provides a set of analog channels. These channels may have some use for simulated sensor based robot programming. For example, in "Verify"-mode you can set these values just before the command line is executed, where the ANAIO-channel is used by the IRDATA-program. You can see if the program performs as planned. The required float-value is entered in the textport. The display is:

```
Enter Channel-Number [0..9]: <c>
Current value  [c]: <old_value>
Enter FLOAT -Value: <input_value>
```

- **Set DIGIO:**

  Is used to set digital I/O-channels to a value. The virtual robot controller provides a set of digital channels. These channels may have some use for simulated sensor based robot programming. For example, in "Verify"-mode you can set these values just before the command line is executed, where the DIGIO-channel is used by the IRDATA-program. You can see if the program performs as planned. The required 32-bit-value is entered in the textport. The display is:

```
Enter Channel-Number [0..9]: <c>
Current value  [c]: <old_value>
Enter INT-Value: <input_value>
```

- **Exe. POSITION Dataset**

  A file of position data is read and executed (not a robot program). This facility provides a simple interface from other programs for joint position time series input to KISMET. Thus, KISMET can be used for postprocessing.

10.2.2 Verify-Submenu

The "Verify"-mode is used for off-line simulation of robot programs in single step mode (verify mode).

Note: All commmands marked with ‚{}‘ are executable only for robot languages (codes) which allow no arithmetic commands, nor conditional program execution statements (program flow control). These KISMET-commands cannot be used for IRDATA- or ICR-programs. The use of all commands marked with ‚{}‘ is limited for simple AD HOC programmed robot codes.

The Verify Submenu is a so called Pop-Up Menu:

Step forward
Step backward
Position printout
Jump to Start
Jump to End
Modify Position

Insert Position
Delete Position
Start Repeat
Teach Append
Overwrite Teachfile
Verify off
Activate Teachfile

Figure 29: The Verify Submenu

- **Step forward**

  Executes the next program step.

- **Step backward {}**

  Executes the previous program step. This function allows program traversal in reverse order.

- **Position printout**

  The current robot parameters (joint- and TCP-positions etc.) are printed out.

- **Jump to Start {}**

  The program counter is set to the first command line (start step). Step counter and initial position are reset.

- **Jump to End {}**

  The program counter is set to the last command line (end step).

- **Modify Position {}**

  This command is used for teachfile modification. The current position of the active robot is used as motion target for the actual program step. You may move the active robot at any time (step) during teachfile verification.

- **Insert Position {}**

  The current position of the active robot is inserted as an additional step for the next motion target in the robot program. This function is used to enter intermediate position in the robot path.

- **Delete Position {}**

  This command will delete the current program step.

- **Start Repeat**

  Stops "Verify"-mode (single step execution). The robot program will continue execution in continous execution mode ("Repeat"-mode).

- **Teach Append {}**

  Allows to append additional program steps at the end of the robot program. This function works similar to "Teach"-mode. KISMET program flow now continues in "Teach"-mode.

The program counter is set to the new end position (last step). The number of program steps is incremented by one.

- **Overwrite Teachfile**

  The modified robot program is saved (written to the file system). KISMET will request the filename. If the original filename of the robot program is used, this file will be overwritten, otherwise a new file is created.

- **Verify off**

  Stops "Verify"-mode.

- **Activate Teachfile**

  This function activates "Verify"-mode (single step execution) for another robot teachfile (program). The command allows especially to test multiple teachfiles which are loaded in KISMET for different robots. This means, it is possible to simulate in workcells "Verify"-mode with more than one robot.

10.2.3 TCP-Track Display

- **TCP-Display ON/OFF**

  Enables/disables the display of the TCP-TRACK. This function is used to vusualize the movement of a tool, for example in robot programs. Points are entered ino an internal list each time when the TCP has moved a user definable distance. For display, the points are connected by lines. The display type and the distance can be defined by using the "Display MODE" button and the 2 "SET" buttons. Figure 30 shows a typical example of the TCP-path as displayed in KISMET.

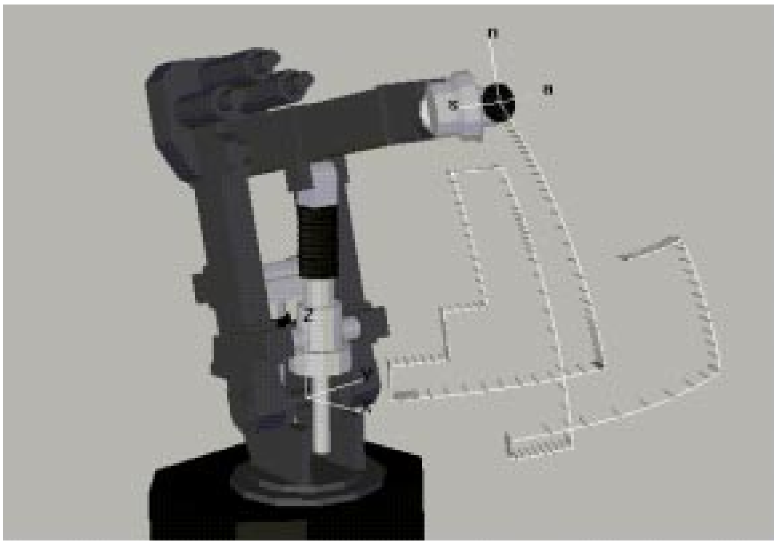


Figure 30: Display of TCP-path

- **Display MODE**

  Activates a submenu in the KISMET kernel with display modes for the TCP-track. The various modes are:

  - TRACK only
  - TRACK + a-normal mode 1
  - TRACK + a-normal mode 2
  - TRACK + TAGs
  - Point mode
  - Spline TRACK

- **SET TAG Leng**

  is used to define the length of the coordinate axes of the TAG frame in object space. The default value is 50 mm.

- **SET PNT Dist**

  This command is used to set the minimum distance for entering control points in the TRACK list. When the TCP is moved from the last position in the list for at least the defined distance, another point is entered in the list. The default value is 50 mm.

11 TEACH

11.1. Introduction to "TEACH"

Starts and Stops the "Teach-mode". Teach-mode is used for robot program generation, using simple geometric teaching. This is, the robot is first moved to the next target position in KISMET. When the user is satisfied with the position, in the second step the robot motion command is generated for this position in the robot program. According to the current robot programming code in use, various submenus are displayed by KISMET.

Parameters for some commands are requested by KISMET interactively in the textport. When you start teaching, you are asked by KISMET to enter the program name of the file to be created. The string you enter at this point is used for the comment in the title line of IRDATA only. The final filename is entered in the textport after stopping the TEACH-mode.

During teaching, the generated code is stored locally (in main memory) by KISMET. The final code is saved on disk only after you have stopped teaching.

Note: The full documentation for the different robot programming codes is not given in this guide. Available programming codes and commands are partly installation dependent.

11.2. The TEACH Panel

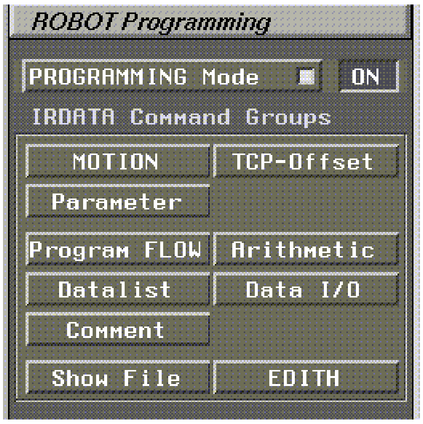


Figure 31: The TEACH Panel

- **Programming Mode**

  Toggles Programming mode. When switched to "ON", KISMET asks for a file name, the string entered at this point will be used in the header section of the robot program to be generated. After the programming is finished, switching to the "OFF" state will store the program as a file. KISMET will ask at this point again for the filename.

- **Motion**

  will produce a menu with available motion commands, e.g. PTP-motion, LINEAR-motion, JOINT-motion etc.

- **TCP-Offset**

  A submenu with available commands for TCP- and ZP-offset definition will be shown

- **Program Flow**

  Shows a submenu with available commands for program flow control within the robot program.

- **Arithmetik**

  This button shows a submenu with available commands for arithmetics in the robot program. In IRDATA a powerful set of arithmetic commands is available.

- **Datalist**

  Shows a list of robot ptogramming commands for datalist handling by the robot program

- **Data I/O**

  Gives available robot programming commands for analogue and digital I/O operations in the robot program.

- **Comment**

  Is used to enter a comment line in the robot program

- **Show File**

  This command will show in a subwindow an ASCII printout of the robot program as generated so far.

- **EDITH**

  A specific command set (an extension to IRDATA) will be shown as specifically defined for the kinematically redundant EDITH robot.

12 WFRM

12.1. Introduction to WFRM

The "Edit WFRM" submenu is used for creation, editing and positioning of "Workframes" (WFRM), WFRM-PATH objects, and WFRM-PATH files.

A WFRM is the supertype datastructure, which is used in KISMET to define TCP reference positions on the objects to be handled. The basic task in robot programming is to move the robot TCP in a specified way from one WFRM to the next and perform at these positions a specific task. To automate or simplify off-line program generation, KISMET allows the definition of additional data with each workframe. This is a name (identifier), a comment, a type-id, a visibility number etc.. The WFRM is connected to a FRAME through a transformation matrix, which allows local, relative positioning. A group of WFRMs in a specific, user-defined order forms a WFRM-PATH. A set of WFRM-PATH structures is stored in a WFRM-PATH-FILE.

WFRM-data can be passed during KISMET program execution to other programs through a file. A SCRIPT command is used to initiate this action.

Additionally to the standard function of a WFRM, as described above, the WFRM data-structure is used in KISMET to define CAMERAs, moveable scene LIGHTS, USS-sensors (simulated ultrasonic sensors), robot TCP- and ZP-systems. For identification of the WFRM-class and possible operations, a type attribute is associated with each WFRM.

12.2. The WFRM Panel

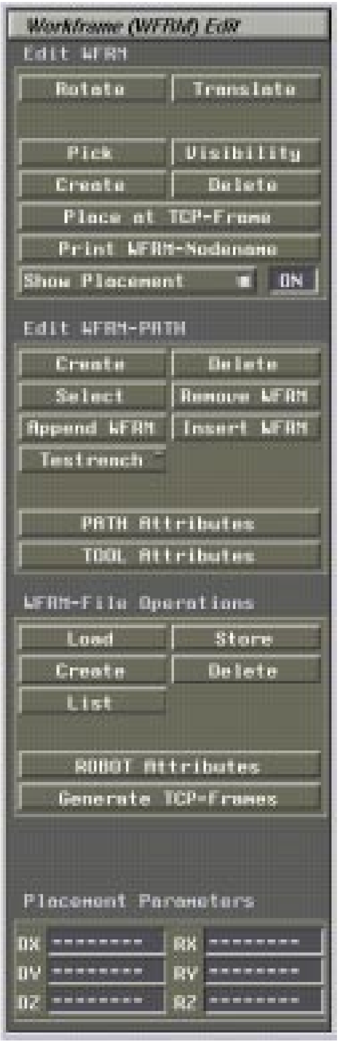


Figure 32: The WFRM Panel

12.2.1 The Edit WFRM Commands

- **Rotate WFRM**

  The WFRM-object is rotated in 3 dimensions relative to its reference FRAME. A general 6 parameter transformation (*dx* , *dy* , *dz* ,α,β,γ) is used by KISMET for the transformation.

  - **LM : Change α_G, i.e. rotate WFRM around X-axis**
  - **MM : Change β_G , i.e. rotate WFRM around Y-axis**
  - **BM : Change γ_G, i.e. rotate WFRM around Z-axis**

- **Translate WFRM**

  The WFRM-object is translated relative to its reference-FRAME in 3 dimensions.

  - **LM : Translate WFRM about d_x along FRAME-X-axis**
  - **MM : Translate WFRM about d_y along FRAME-Y-axis**
  - **BM : Translate WFRM about d_z along FRAME-Z-axis**

- **Pick WFRM**

  This command is used to start a PICK operation to identify the WFRM, WFRM-PATH, or WFRM-PATH-file to be edited. A WFRM is picked at the origin of its coordinate system as drawn in the workcell display.

- **Visibility**

  Allows for each WFRM to define its drawing mode individually. The display options are as follows:

invisible
show FRAME
show Text
show FRAME + TEXT

  In invisible mode nothing at all is displayed, except the picking region in WFRM-Pick mode. FRAME is used to display the coordinate frame of the WFRM, TEXT displays the WFRM-identifier (name).

- **Create**

  This command is used for interactive creation of a new WFRM object. After the command is started you have to pick a reference FRAME to which the new WFRM is connected in the model-tree via a PLACEMENT transformation. KISMET will ask you for the type of the new WFRM-object. The type is defined by the type-id, which is stored as an integer number in the model. The type is used also to identify any possible functions or operations to be possible with that WFRM in later stages of the simulation. For example if you chose the CAMERA-type, a camera-view will be associated with the WFRM and it can be used to perform CAMERA simulation.

  WFRM types are:

  - **OBJECT-WFRM**

    type [0...9999] These are WFRMs which can be used in WFRM-Path definitions. They are potential target positions for the ROBOT-TCP on the workcell objects. These WFRMs move with the parts to be manipulated.

  - **TCP**

    type [10000...10999] The WFRM is a ROBOT tool-centre-point frame (TCP). The TCP shall be attached to a ROBOT-mechanism only. It should be attached to hand coordinate system. The placement (TCP-offset) is later moved into the workpoint of the current tool.

  - **ZP**

    type [11000...11999] The WFRM is a ROBOT base reference system or zeropoint (ZP) coordinate system. It defines the reference frame for robot motion targets in cartesian coordinates. The placement can be changed using the "Change ZP" commands.

  - **CAMERA**

  type [15000...15999] This defines a CAMERA-VIEW. It can be used for CAMERA simulation. The name of the WFRM is used in the "Select CAMERA" submenu for identification of the current CAMERA.

  - **USS**

    type [16000...16999] Defines a simulated ultrasonic sensor (or similar objects).

  - **LIGHTS**

    type [17000...17999] This type is used to attach a lightsource (defined in the '.lgt' file) to a part. When the part is moved in the workcell, the light will be moved as well. The lightsource number [0..7] is entered in the textport. You may only use as many sources as defined in the '.lgt' file

- **Delete**

  The picked WFRM is erased from the runtime model datastructure.

- **Place at TCP-Frame**

  The WFRM-placement is recalculated. The WFRM will be placed at the current location and orientation of the active ROBOTs TCP.

- **Print WFRM-Nodename**

  The full qualified nodename of the WFRM reference FRAME and the WFRM-name are printed in the textport. These names can be used in SCRIPT-commands related to WFRM operations. Note: In SCRIPT-commands, WFRM references require the nodename(s) of the reference FRAME and the WFRM-Id, i.e. two strings.

- **Show Placement ON/OFF**

  Displays the 'Placement-Parameters'.

12.2.2 The Edit WFRM-Path Commands

- **Create WFRM-Path**

  This function is used to create a new WFRM-PATH. The attributes of the new PATH are entered interactively. Following parameters are to be defined by the user:

```
Enter WFRM-PATH name : <PATH-NAME>
Enter FRAME colour [0..127] : <ci>
Enter TEXT colour [0..127] : <ci>
Enter NUM colour [0..127] : <ci>
```

  The data to be entered are the <PATH-NAME> (string), and the colour indices <ci> (integer) used by KISMET for FRAME, TEXT and NUM drawing, respectively. The colour indices refer to the constant c'olour table as defined in the '.col'-file.

- **DELETE**

  The selected WFRM-PATH is removed from the runtime datastructure and unlinked from the PATH-FILE.

- **SELECT**

This command is used to specify a WFRM-Path for editing or simulation. The PATH is selected by PICKING (LM) a WFRM which is already connected to the PATH. All WFRMs of the PATH are highlighted after selection.

- **REMOVE WFRM**

  This command is used to REMOVE a WFRM from the WFRM-PATH datastructure. The WFRM is unlinked from the PATH only. It will remain in the runtime model, linked to the reference FRAME as an "ordinary" workframe. You can relink the WFRM at another position in the same PATH or another PATH. The term ’position’ refers here to the execution order of the WFRMs in the PATH. To delete the WFRM completely from the model, use additionally the "Delete WFRM" command.

- **Append WFRM**

  The selected (picked) WFRM is linked to a WFRM-PATH at the current end position of the PATH.

- **Insert WFRM**

  The selected (picked) WFRM is inserted into the WFRM-PATH at a user-defineable position. You have to pick another WFRM already connected to the PATH after executing this command. The active (selected) WFRM will be entered before the position of the picked WFRM.

- **Testreach**

  This command allows stepping through the PATH, using the selected PATH and the active ROBOT. The robot TCP is moved to each WFRM in the PATH. You can use this command for reachability and collision tests. The WFRMs are processed according to their order in the PATH. You can move in both directions (regular and reverse order) through the PATH. When in STEP-mode, a submenu is called for each step. It allows the commands:

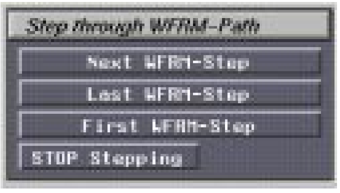


Figure 33: The Testreach Panel

  - **Next WFRM-Step**

    The active ROBOTs TCP is moved to the next WFRM position in the PATH.

  - **Last WFRM-Step**

    The active ROBOTs TCP is moved to the previous WFRM position in the PATH.

  - **First WFRM-Step**

    The active ROBOTs TCP is moved to the first WFRM position in the PATH, i.e. to the start WFRM.

  - **STOP Stepping**

    STEP-mode is stopped.

- **PATH Attributes**

  This command is used to modify the PATH attributes interactively. Following parameters are to be defined by the user:

  ```
  Enter WFRM-PATH name : <PATH-NAME>
  Enter FRAME colour [0..127] : <ci>
  Enter TEXT colour [0..127] : <ci>
  Enter NUM colour [0..127] : <ci>
  ```

  The data to be entered are the <PATH-NAME> (string), and the colour indices <ci> used by KISMET for FRAME, TEXT and NUM drawing, respectively. The colour indices refer to the constant colour table as defined in the ’.col’-file.

- **TOOL Attributes**

  The actual robot TCP-offset (active ROBOT) is entered into the WFRM-PATH data-structure, modifying the parameters of the ’TOOL’ command. This function is useful when you try several tools with different TCP offsets for the current PATH.

12.2.3 WFRM-File Operations

- **Load**

  This command is used to READ an existing PATH-FILE into the workcell for editing or simulation. Pick the PATH-filename from the file panel. All files with the suffix ’.wpf’ or ’.WPF’ are displayed for selection. Note: PATH-FILEs are stored in the directory ’$kis_home/teachlib’.

- **Store**

  Is used to write a PATH-FILE to the filesystem. PATH-FILEs are stored in the directory ’$kis_home/teachlib’. You should use the filename suffix ’.wpf’ or ’.WPF’.

- **Create**

  This command is used to create a new PATH-FILE and to initialize its attribute data. The following parameters are to be defined by the user by using the textport:

  ```
  Enter WFRM-PATH filename : <filename>
  Enter COMMENT Line : <comment string>
  ```

  Note: The predefined suffixes for WFRM-PATH-files are ’.wpf’ or ’.WPF’.

- **Delete**

  The selected PATH-FILE with all linked WFRMs is deleted from the runtime datastructure (KISMET main memory).

- **List**

  The selected (active) PATH-FILE is listed in an overlay text-window.

- **ROBOT Attributes**

The actual ROBOT parameters are entered into the WFRM-PATH data-structure, modifying the parameters of the ’ROBOT’ command. This function is useful when you examine several robots with the current PATH.

- **GENERATE TCP-FRAMES**

  This function computes for all WFRMs in the current PATH-FILE the TCP coordinates in robot ZP-space. To select a PATH-FILE you have to select one of the WFRM-PATHs, using the "SELECT WFRM-Path" command. The active ROBOT and the selected PATH-FILE are used for the computation. When a position cannot be reached the calculation is stopped. You should then change the robots base position in the workcell and execute "GENERATE" again. Repeat this procedure until all WFRMs in the PATH-file can be reached collision free. Then save the PATH-FILE. The resulting PATH-FILE can be used by external modules for automatic robot program generation.

  This command is most useful after editing relative WFRM positions, or moving FRAMEs in the workcell model. For recalculation of the TCP positions in ROBOT coordinates, simply execute the "GENERATE" function.

12.3. The WORKFRAME-Pathlist Concept in KISMET

The main problem in off-line-programming is the combination of geometry- and topology-data of the robot workcell, as available in the Computer Aided Robotics-(CAR) system (KISMET), with textual programming features. Whereas the structured geometry information. Another possible expression could be the term "geometric knowledge base" of the working environment, as available in the CAR-system, can be modified after changes in the real workcell rapidly (using the CAD editing functions), in the case of pure textual programming alle target coordinates which are affected by the change in the workcell would have to be edited or reentered manually. This is a time consuming task. It is a common procedure in pure textual off-line programming, to enter the real target positions of the motion commands into the textual program in another workstep. This step is called "teaching". An operator has to move the real robot "On-Line" in the real workcell manually from position to position. The actual kinematic data is then transferred from the robot controller into the textual program at each target point. After any changes in the real cell, this step is to be repeated again. This concept is called hybrid programming. Most of the advantages of off-line programming are lost in this concept.

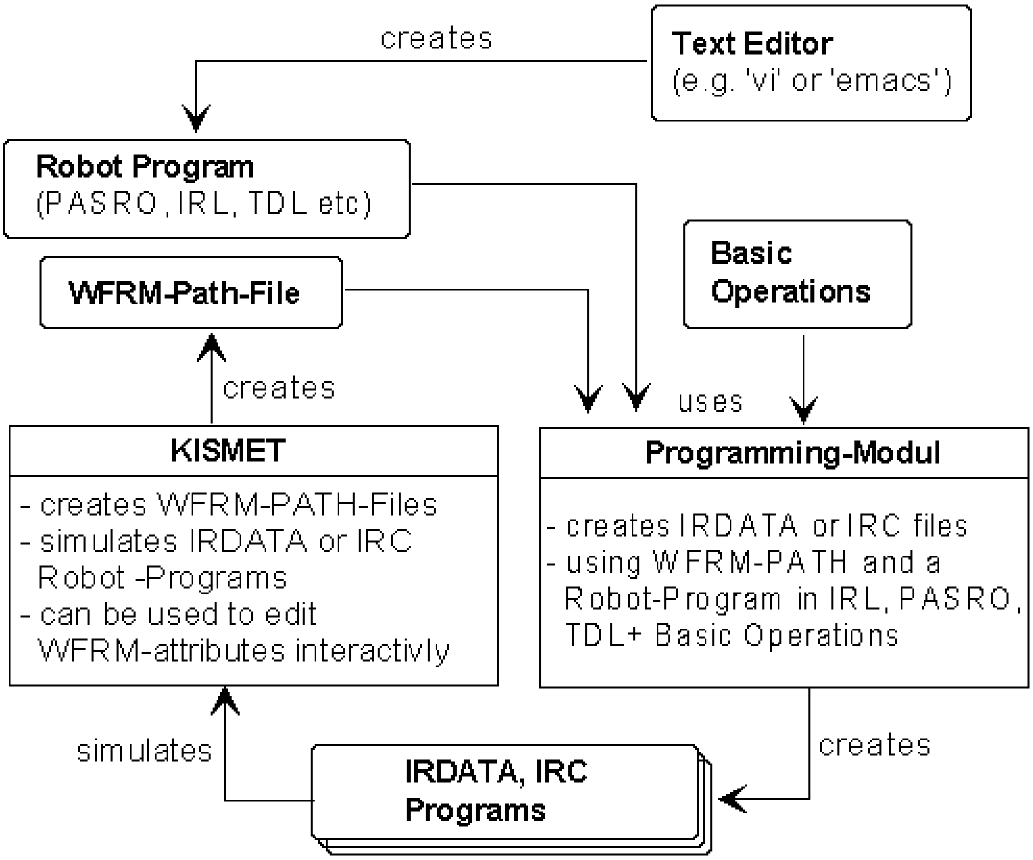

creates
Text Editor
(e.g. 'vi' or 'emacs')
Robot Program
(PASRO, IRL, TDL etc)
WFRM-Path-File
Basic Operations
creates
uses
KISMET
- creates WFRM-PATH-Files
- simulates IRDATA or IRC Robot -Programs
- can be used to edit WFRM-attributes interactivly
Programming-Modul
- creates IRDATA or IRC files
- using WFRM-PATH and a Robot-Program in IRL, PASRO, TDL+ Basic Operations
simulates
creates
IRDATA, IRC Programs

Figure 34: WFRM-Path Lists- Module architecture and dataflow in the OLP-system

Using KISMET and the workframe concept, as described in this section, is a combination of graphical- and textual programming techniques, this time consuming step, as described above, will become obsolete.

The workframe concept is mainly intended to automate the transfer of motion command target positions into the textual program body, especially after any changes in the real workcell geometry.

In this concept, all motion targets used in the high-level robot programming language (e.g. PASRO, ROBCAD-TDL, VAL-II, IRL etc.) are specified using symbolic adressing. All working positions of a robot task (workframes: WFRM) are define in the structured geometric database of KISMET. The WFRMs are connected via references to the objects in the workcell. A transformation matrix allows positioning of the WFRM relative to the object. Thus, the workcell assembly topology is also connected with the WFRM definitions. All WFRMs of a subtask are combined into a WFRM-PATH. The sequential order of the WFRMs in the PATH corresponds to the order of the target positions in the final robot program. Each WFRM-PATH can be uniquely identified, using the WFRM-Id textstrings, which include the assembly topology and the WFRM name. All WFRM-PATHs of a robot or handling task are combined to a WFRM-PATH sequence and stored in the filesystem as a WFRM-PATH-FILE (physical representation).

An off-line programming module (OLP-module), which is specific for the high level programming language used, and for the executable robot target code (IRDATA, ICR) to be generated, compiles from the WFRM-PATH files and the high-level program body, together with a library of elementary operations, the final executable robot program.

In the WFRM-PATH file the position of each WFRM is redundantly stored. This is

- relative to the robot reference system (ZP-system). These informations (robot TCP-offset, position of additional axes, hand status etc.) are required from the OLP-module. They can be transferred directly into the IRDATA program. The identification of the WFRM-data in the WFRM-PATH file is done with the help of symbolic names, i.e. the PATH-name and/or the single WFRM-Id. These names have to be used in the high-level body either directly, or a conversion table is to be used.
- relative to the workcell object to be handled in the geometric database with its relative transformation to the Object-FRAME. The object-ID (FRAME-ID) is stored together with the other data in the WFRM-PATH-FILE. The WFRM-ID is built from the IDs of the corresponding object-FRAME as a full qualified string, for example:
  JET_TORUS.OCTANT_5.UPPER_LIMITERS.LIMITER_2.COOLING_PIPE
- This concept permits especially any geometric editing of existing WFRM-PATH lists in KISMET after any changes in the real workcell geometry.

After any position change of a single WFRM in the geometry database of KISMET, another processing step is used to compute all WFRM positions relative to the robot base or ZP-system, which is required by the robot controller. A powerful command, called GENERATE ROBOT-Frames, is available in KISMET to perform this task. The PATH-FILE is stored back with the now modified target positions. The PATH-FILE is passed to the OLP-module.

Finally, the OLP-module generates with an already existing high-level program body and the KISMET modified, corrected PATH-FILE again an executable IRDATA-program (or any other programming code). Before download, the IRDATA-program can be simulated in KISMET using the "Repeat" submenu.

Another editing of the high-level program is not necessary using this concept.