ACTION: A data record describing an action to be dispatched by the dispatcher to a server. In particular an action contains (or points to) a scheduling record which describes when and where to do the action and a task record which describes the action to perform. For example each data acquisition device in the tree includes actions which sequentially schedule ARM and STORE tasks during the shot sequence.

AXIS: A set of data types used to represent functions which describe independent parameters. For example a single speed clock data type could be stored as a single delta-time representing an infinitely long array of times when the clock pulsed. This data type is referenced in a dimension.

BROTHER: All of the children of a node are brothers. The list of brothers is a singly linked ordered list. Each brother has an identical parent pointer pointing to the parent.

CHARACTERISTIC: Each node has a list of characteristics which describe the data pointed to by the node. These include things like ON/OFF status, data location, data type, data validation, etc... The characteristics of the nodes for a given shot are stored in their own file. They can be kept on line for much longer than the data (potentially indefinitely) because these files are relatively small. All of the characteristics are readable using the TREE$GETNCI routine with item codes defined in NCIDEF.H. Some of the characteristics are setable using TREE$SETNCI and others are filled in by TREE$PUT_RECORD etc...

CHILD: Node in the tree has zero or more children which are in turn also nodes. The children of a node are a singly linked ordered list (optionally sorted).

CONGLOMERATE: A conglomerate is a collection of contiguous nodes. Once a node is added to a tree its location in the tree file remains constant, the links stored in the node change with the node's position in the hierarchy. conglomerates are used to implement CAMAC model support. When a device is added to the experiment model a collection of nodes is added to the tree with some device dependant default structure. When the device support code is called it references the the individual pieces of information about the device using the conglomerate index not using the tree structure. This means that the nodes which describe a device can be moved anywhere in the hierarchy with out impacting the device support code.

DATA: Used in different contexts:
- In the usual sense the information about what happened during a shot of the experiment.
- The part of the experimental model which contains the actual numbers acquired from the A to Ds is the data part of the model.
Glossary of terms for MDS-Plus

**DESCRIPTOR**: A data structure which describes a piece of data being passed around. In general MDS-Plus supports the VMS descriptors for scalers, strings, and arrays and an extended descriptor for describing large chunks of bytes.

**DEVICE**: A data acquisition data source like a CAMAC waveform digitizer. In order to use MDS-Plus to acquire data from the experiment, place devices which describe your data acquisition hardware into the tree.

**DIMENSION**: Description of an independent parameter. A dimension relates a window with an axis to define the real values which correspond to one dimension of an array data item. The window describes the piece of the potentially infinite independent axis the data was sampled at. For example for a channel of a digitizer with a single speed clock the axis would be the sampling frequency and the window would contain the trigger time, and the number of samples.

**DISPATCHER**: A program which will schedule all of the automatic Data Acquisition and Data Analysis tasks. This program will determine what needs to be done and broadcast messages to servers which will actually to the work. Dispatching is cluster wide and the messages are passed to the servers using the VMS lock management services.

**EXPERIMENT MODEL**: The tree is used to model an experiment hence is called the experiment model.

**LINDA**: Acronym for "Linguaggio di Descrizione per Acquisizione dati" which translates to "description language for data acquisition." The linda language is used by the physicists to define the experiment model.

**MEMBER**: Each node of the tree can have zero or more members. These are nodes which are used to store a list of named values which are associated with the node. Colons in a path name are used for the membership relationship. For example:

```
EXPERIMENT.DIAGNOSTIC.MEASUREMENT
```

might have the MEMBERS COUNTS, TIMING, ENGINEERING. These nodes, which actually contain the corresponding data are called:

```
EXPERIMENT.DIAGNOSTIC.MEASUREMENT:COUNTS
EXPERIMENT.DIAGNOSTIC.MEASUREMENT:TIMING
EXPERIMENT.DIAGNOSTIC.MEASUREMENT:ENGINEERING
```

**METHOD**: Something which can be done to a node (a la smalltalk). For example, if a node is part of a conglomerate then you can store and initialize it. If it is a signal you could draw it.

**NODE**:  
1. A node is the basic primitive element of storage in the MDS-Plus system. Each node represents a position in the hierarchy. Nodes may contain a data item (of any type) and have zero or more children and zero or more members. In addition, each node has a list of characteristics which describe the contents of the node.
2 A node is only the part of (1) which has to do with the structure of the tree. The tree is stored in three pieces:

- **NODES** - Tree structure information, like brothers, parents, members, and readonly characteristics.
- **CHARACTERISTICS** - READ/WRITE characteristics like ON/OFF status, Data validity, Data location, etc...
- **DATA** - Data values associated with the nodes (1).

**PARAMETER:** A parameter is a structured data type which is used to associate a data item with a validity specification and a help string. These are used to represent knobs which users will be able to interactively change in the model. Some examples of parameters:

```
...MEASUREMENT:CALIBRATION
...MEASUREMENT:POSITION
...MEASUREMENT:SETTING
```

**PARAMETER EDITOR:** The parameter editor is an interactive program used to modify the values of parameters. This editor will display the associated help strings and check the values against their validity specifications.

**PARENT:** Every node in the tree, with the exception of the root node, has a parent. All of the children and members of a node have that node as their parent.

**PATH:** A path is a specification of a node in a tree. It consists of a series of names separated by delimiter where each name corresponds to one level of the tree. ‘.’s are used to specify child/son relationships and ‘:’s are used for member relationships. Optionally a path can be prefixed by a tag name which is a name attached to some particular node in the tree. Some sample paths:

```
EXPERIMENT.DIAGNOSTIC.MODULE_1
EXPERIMENT.DIAGNOSTIC.MODULE_1:GAIN
EXPERIMENT.DIAGNOSTIC.MODULE_1:OFFSET
EXPERIMENT.DIAGNOSTIC.MODULE_1:COUNTS
\TIMING::MY_TRIGGER
\TIMING::MY_TRIGGER.SUB_TRIG_1
```

In many instances wildcards can be used when specifying a path.

- ‘*’ - like in VMS for any name or part of a name. This wildcard does not cross tree level daries.
- ‘%’ - like in VMS substitutes for one character in a name.
- ‘...’ - like in VMS matches all of the children of a node.
- ‘***’ - similar to ‘...’ above but substitutes for all sons and members of a node.
- ‘^’ - Used to find an ancestor (PARENT, GRANDPARENT, etc...) ^GUB - some ancestor named GUB.

**PATH NAME:** See path.

**ROOT NODE:** A special node which is at the top of the tree. This node is created when a new tree is opened. It is called ‘TOP’ and given a tag \ TOP’.)
Glossary of terms for MDS-Plus

**SCHEDULING:** An action is made up of a task and scheduling record. The scheduling defines when the task should or can be performed and which server(s) can execute the task. There are three types of scheduling envisioned so far:

1. **SEQUENTIAL** - Each action is performed in a specified order for some phase of the experiment. This mechanism will be used for the bulk of the data acquisition tasks.
2. **CONDITIONAL** - The actions are performed when a condition is met during a phase. ECS from PPPL uses this form of scheduling.
3. **ASYNCHRONOUS** - Actions which are performed in response to an asynchronous event such as a CAMAC LAM, device interrupt, or MDS EVENT.

**SERVER:** A program which receives messages from the dispatcher and executes the specified tasks.

**SIGNAL:** A signal is a structured data type which is used to represent arrays with associated independent parameters. Signals have two parts:

1. **DATA** - An N-dimensional array of data values.
2. **DIMENSIONS** - For each dimension of the data a description of the related independent parameter.

The most common application of signals is the output of transient digitizers. In this case the data is a vector of values from the digitizer and their is one dimension which is a description of the times at which samples were digitized.

**SHOTID:** A shotid or shot_id is a unique identification of an instance of the experiment. Typically the shotid is made from a unique shot number and or the date on which the shot took place.

**SUBTREE:** See tree.

**TAG:** Each node in a tree can have a list of tags associated with it. These are names which can be used by applications and referenced by other nodes which are independent of the hierarchy. Tags can be used to define a subset of the information in a tree or subtree which is of general interest to users of the system.

**TASK:** A task is a structured data type which describes an operation to be performed by a server. There are three types of tasks:

1. Routines which are called directly by the server.
2. Programs which are run (or poked) by the server.
3. Command procedures for an interpreter to execute (DCL, IDL, etc...)
4. Methods which are done on nodes in the tree. (See method above.)

**TREE:** A tree is a collection of files which contain the experiment model. The model is made up of nodes which are either links to subtrees or contain the data and linkages directly. A tree is: A tree and all of its subtrees.
**WINDOW:** A data type which is used to describe a slice of an axis. A window consists of:

- **XREF** - a reference time
- **NBEFORE** - the number of samples before the reference time
- **NAFTER** - the number of samples after the reference time