MtransformationMatrix:

An MTransformationMatrix allows the manipulation of the individual transformation components (eg scale, rotation, shear, etc) of a four by four transformation matrix.

The transformation in the node is represented as a 4x4 transformation matrix.

This class allows access to the whole matrix, or the individual components (eg scale, rotation, shear, etc) of the transformation. This breakdown provides animators fine control over the animation of these parameters. Therefore, it is necessary to describe the order in which these attributes are applied to build the final matrix attribute.

A transformation matrix is composed of the following components:

\li Scale pivot point point around which scales are performed [Sp]

\li Scale scaling about x, y, z axes [S]

\li Shear shearing in xy, xz, yx [Sh]

\li Scale pivot translation translation introduced to preserve existing scale transformations when moving pivot. This is used to prevent the object from moving when the objects pivot point is not at the origin and a non-unit scale is applied to the object [St].

\li Rotate pivot point point about which rotations are performed [Rp]

\li Rotation orientation rotation to orient local rotation space [Ro]

li Rotation rotation [R]

\li Rotate pivot translation translation introduced to preserve exisiting rotate transformations when moving pivot. This is used to prevent the object from moving when the objects pivot point is not at the origin and the pivot is moved. [Rt]

\li Translate translation in x, y, z axes [T]

Note that the default RotationOrder is kXYZ.

The matrices are post-multiplied in Maya. For example, to transform a point P from object-space to world-space (P') you would need to post-multiply by the worldMatrix. (P' = P x WM)

The transformation matrix is then constructed as follows:

```
\code
    -1 -1
    [Sp]x[S]x[Sh]x[Sp]x[St]x[Rp]x[Ro]x[R]x[Rp]x[Rt]x[T]
\endcode
```

where 'x' denotes matrix multiplication and '-1' denotes matrix inversion

\code

Sp	= 	1 0 0 spx	0 1 0 spy	0 0 1 spz	0 0 0 1	St	=	1 0 0 0 0 1 0 0 0 0 1 0 sptx spty sptz 1
S	= 	sx 0 0 0	0 sy 0 0	0 0 sz 0	0 0 0 1	Sh	=	1 0 0 0 shxy 1 0 0 shxz shyz 1 0 0 0 0 1
Rp	= 	1 0 0 rpx	0 1 0 rpy	0 0 1 rpz	0 0 0 1	Rt	=	1 0 0 0 0 1 0 0 0 0 1 0 rptx rpty rptz 1
Ro = AX * AY * AZ								
AX	= 	1 0 0 0	0 cx -sx 0	0 sx cx 0	0 0 0 1	AY	=	cy 0 -sy 0 0 1 0 0 sy 0 cy 0 0 0 0 1
AZ	= 	cz -sz Ø Ø	sz cz Ø Ø	0 0 1 0	0 0 0 1	sx sy sz	= =	<pre>sin(rax), cx = cos(rax) sin(ray), cx = cos(ray) sin(raz), cz = cos(raz)</pre>
R	=	RX *	RY * I	RZ (N	lote: ord	ler	is	determined by rotateOrder)
RX	= 	1 0 0 0	0 cx -sx 0	0 sx cx 0	0 0 0 1	RY	=	cy 0 -sy 0 0 1 0 0 sy 0 cy 0 0 0 0 1
RZ	= 	cz -sz 0 0	sz cz Ø Ø	0 0 1 0	0 0 0 1	sx sy sz	= =	<pre>sin(rx), cx = cos(rx) sin(ry), cx = cos(ry) sin(rz), cz = cos(rz)</pre>
Т	= 	1 0 0 tx	0 1 0 ty	0 0 1 tz	0 0 0 1			