

# Maya

## Internal Structure

CIS682

# Programming Interfaces

- MEL - Maya Embedded Language
  - Scripting language
  - Interpreted
  - Fast prototyping, slow execution
- C++
  - Powerful, fast
  - Class libraries

# Maya Architecture

Graphical User Interface

MEL Command Engine

Dependency Graph

# Dependency Graph (DG)

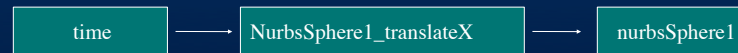
- Data flow model
  - Data manipulated by series of operations
  - Pipeline
  - Push-pull model
- DG - heart of Maya
  - Data and operations represented as nodes
  - Network of nodes to perform task
  - Add functionality by defining new node

# The Scene

- Entire 3D graphics state - the DG
  - Models
  - Animations
  - Textures
  - Lights
  - cameras
- Programming interfaces hide much of DG complexity

# Data Flow

- Nodes
- *Attributes* - properties of a node
- Input/output
- *Compute function*



# NODE

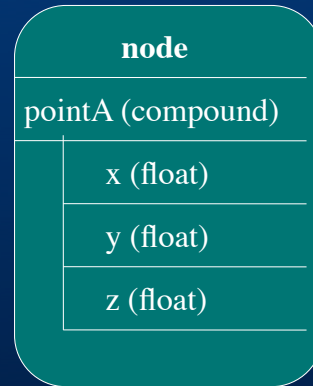
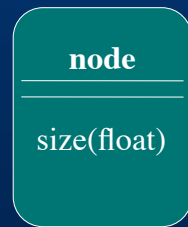
*node*

input

output

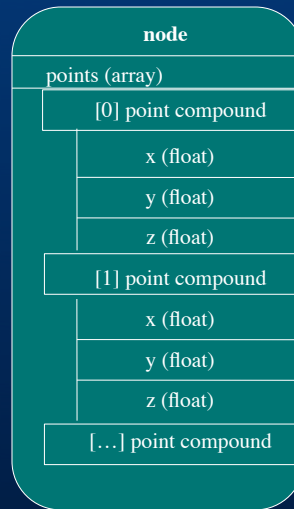
*Compute()*

# Node attributes





# Node



# Node

Custom attributes can be added  
e.g., mass, velocity

Window->Attribute editor  
Attributes->Add Attributes

## Connecting attributes

Window->General Editors->Connection editor

OR

MEL: connectAttr sphere.tx cone.ty

Driven keys as explained by technical group

## Compute Function

- Output = compute(input0, ..., inputN)
- Input and output attributes are LOCAL
- Black box
- Interface: input and output attributes

## Dependent Attributes

- $\text{Volume} = \text{compute}(\text{sphereSize})$
- $\text{attributeAffects}(\text{sphereSize}, \text{volume})$

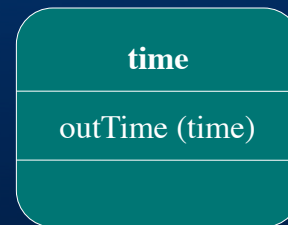
**sphereVolume**

sphereSize (float)

volume (float)

# Time

- Example of node that only holds data
- Current time in time node named time1
- Moving frame slider or click on Play sets time



## Connecting Nodes

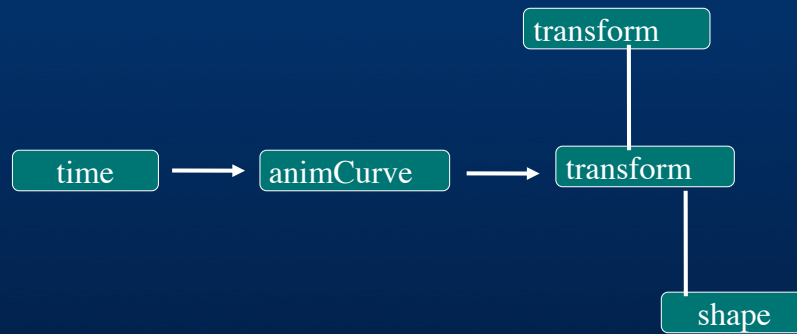
- Connect nodes by connecting node attributes
- Attribute can only connect to attribute of same type
- Maya handles flow of data; node not 'aware' of connections
- Connections: one-to-many mappings
- When connections broken, node retains value

# DAG nodes

- Directed Acyclic Graph
- DAG nodes form Shape-Transform hierarchy
- DAG nodes are in DG - they are DG nodes
  - Some DAG nodes connect to non-DAG DG nodes
  - Some DAG nodes may not be connected to any non-DAG nodes
- Maya shows either DAG hierarchy or connected DG nodes, not both simultaneously



# DAG and DG nodes



## Shape Nodes

- Meshes
- NURBS curves and surfaces
- Springs
- Camera
- Lights
- Particles
- Etc.

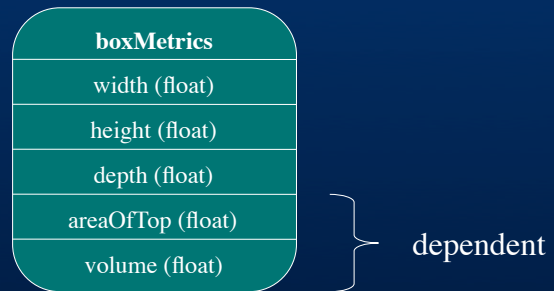
# Transform Nodes

- Shape node can't exist without a transform node
- Shape node holds the data
- Transform node transforms from objects space to world space

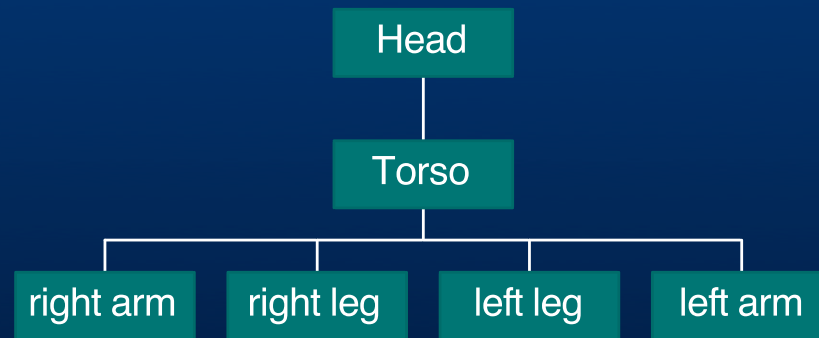


## Dependent Attributes

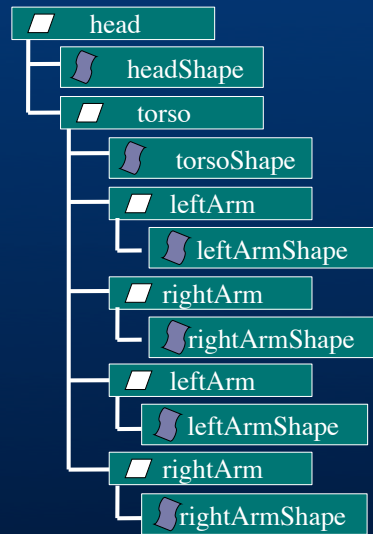
- attributeAffects( width, areaOfTop)
- attributeAffects( depth, areaOfTop)
- attributeAffects( areaOfTop, volume)
- attributeAffects( height, volume)



# Transform Hierarchy



# Node Hierarchy



# Animation

Expressions – expression editor

Baking simulations

Record out put of expressions

Makes each frame a single frame

Connecting attributes

Handling complexity

- Layers
- Groups
- Reference & proxy files