Eurographics 2006

Rémi Arnaud – Mark Barnes – Andrew Lorino
Sony Computer Entertainment

Agenda

- Status and Roadmap
  - COLLADA has become an official industry standard !
  - Goals unchanged since project started
  - 1.4.0 and 1.4.1 release
- Technical overview
  - Geometry
  - Material
  - Textures
  - External references - demo
- Documentation
  - Specification
  - Schema
  - Book
- Tools (live demo)
  - Game Engines
  - Google Earth tools
  - More tools
  - APIs
  - Pipeline tools - demo
- Looking forward
  - Asset management
- Question ?

COLLADA status and roadmap

- Thousands of active users
  - Many professional game developers currently using COLLADA
    - THQ, EA, Konami, NCsoft, DoubleFine, Rockstar …
    - Unreal Engine, C4 engine, Ogre, Irrlicht Engine, …
  - Thousands of downloads every months
- Adopted by Google Earth for 3D models
  - COLLADA content manipulated by mainstream users
  - Many tools targeting Google Earth are adding COLLADA support
    - RealViz, Autocad, SketchUp, Photomodeler
  - 3D models available to COLLADA users
    - http://sketchup.google.com/3dwarehouse/
- More DCC tools have COLLADA support
  - Since COLLADA 1.0: 3dsMax, Maya, XSI
  - Since COLLADA 1.4.x: Blender, Houdini
  - Work in progress: DAZ, Modo, Lightwave
- Official conformance test in development
  - Early access available to Khronos members
  - Publicly available reference viewer and coherency test
- 1.4.x Very stable specification
  - Current focus in quality of implementations
  - 1.4.2 work in progress – targeting March 2007
    - Mainly COLLADA FX additions:
      - Syndication
      - OpenGL ES 2.0 profile

COLLADA is an industry standard

- Not an API, COLLADA is a language
  - Intermediate Digital Asset Exchange format (.dae)
  - XML Schema language technology
- Adopted as industry standard by Khronos since January 2006
- COLLADA 1.4.0 is the first standard specification

KHRONOS GROUP

www.khronos.org/collada

1.4.1

- Service release
  - Primarily fixes issues with complex FX and material bindings
- Released on Khronos web site on June 30th, 2006
- 1.4.0 backward compatible service update
  - 1.4.0 documents will validate against the 1.4.1 schema
  - 1.4.0 applications can import 1.4.1 documents

Design Goals (1)

- Content creation has become increasingly demanding
  - 3D devices expands in complexity and capabilities
  - Size of content ever growing
  - Shrinking production schedules
- No existing standard interchange format
  - Source data in DCC proprietary format
  - No collaboration to create such format
  - Too much effort spent creating exporters
  - Demand is very high from content creators
  - Content creativity limited by lack of this technology
- Import is equally or more important than export
  - Need flexibility creating content pipeline
  - Enable mixing tools for better productivity
  - Opening the door to a variety of third party tools
  - Need a language – import and export required for true communication

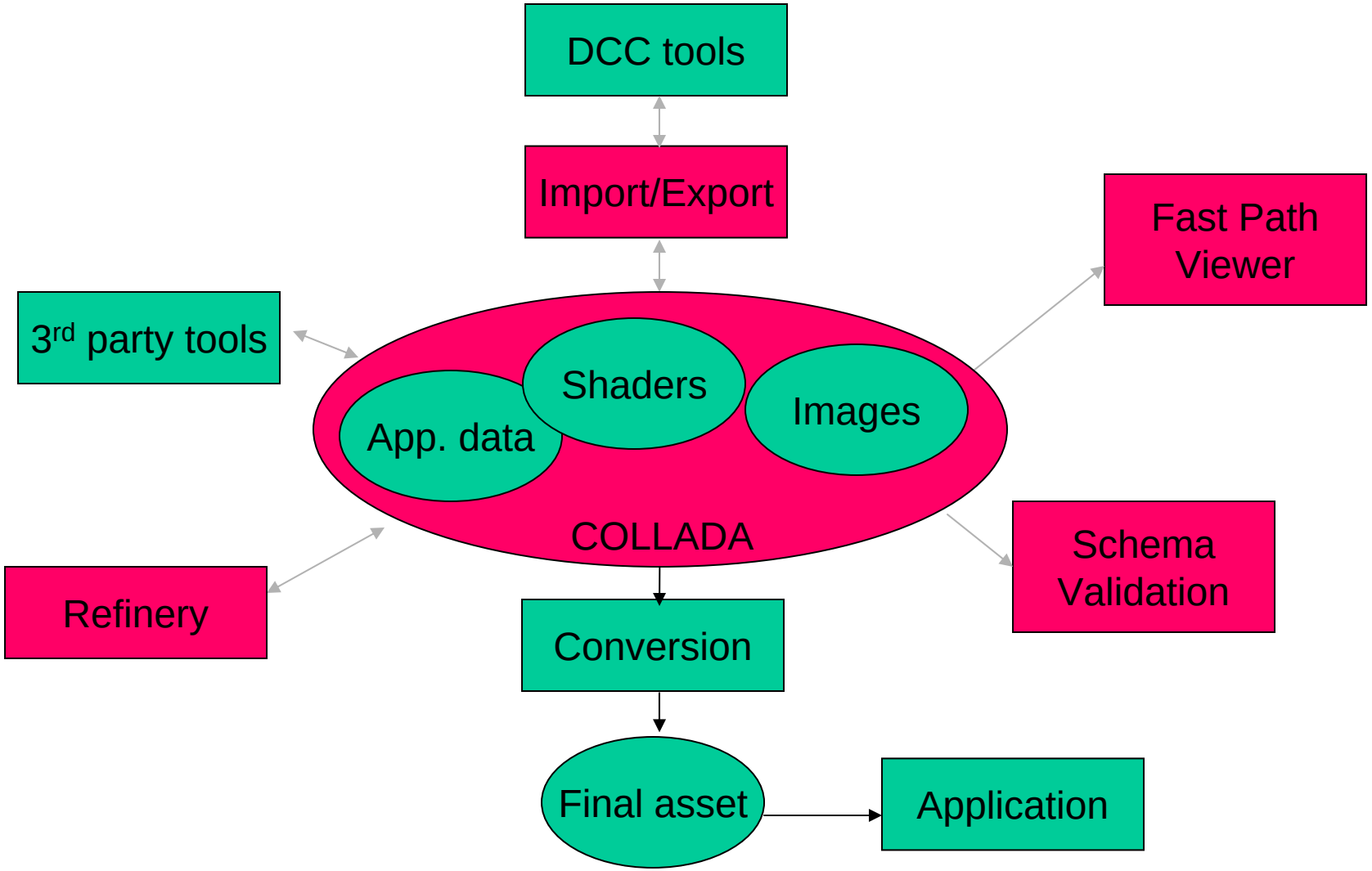
Design Goals (2)

- Targeted for advanced content
  - Need to handle modern features
    - Dynamic content
    - Shader FX
    - Physics
    - … (not limited to graphics)
- Provide a fast-path visualization
  - Direct visualization of COLLADA documents
  - Very fast loading time
- Enable Database Management System
  - Assets are not files, but XML elements
  - Storing the assets in DBMS
- Content pipeline processing
  - Contain enough information to enable creation of target optimized format
  - Can store both DCC centric formatted data and game engine data
    - Polygons with holes
    - Triangle strips

Design Goals (3)

- Extensibility
  - Final user can easily extend the features
    - Extension by substitution
    - Extension by addition
  - Tool vendors can add elements
    - Tool specific data for no-loss import / export
    - Candidate for standardization in future releases
- Modularity
  - Libraries of assets
  - Many things can be externally referenced
- Need validation capability
  - Syntax to be validated, regardless of the content
    - XML Schema validation
  - Need to enable verification of exporter and importer quality
    - COLLADA needs a objective conformance test

COLLADA model



TECHNICAL OVERVIEW

1.4.x Features

- Mesh Geometry
- Transform hierarchy (rotation, translation, shear, scale, matrix)
- Materials
- Textures
- Shader programs (Cg, GLSL, HLSL, GLES)
- Shader effects (FX)
- Lights
- Cameras
- Skinning / bones
- Animation
- Physics (rigid bodies, constraints, rag dolls, collision volumes)
- Instancing
- Techniques
- Multi-representations
- Assets
- User data

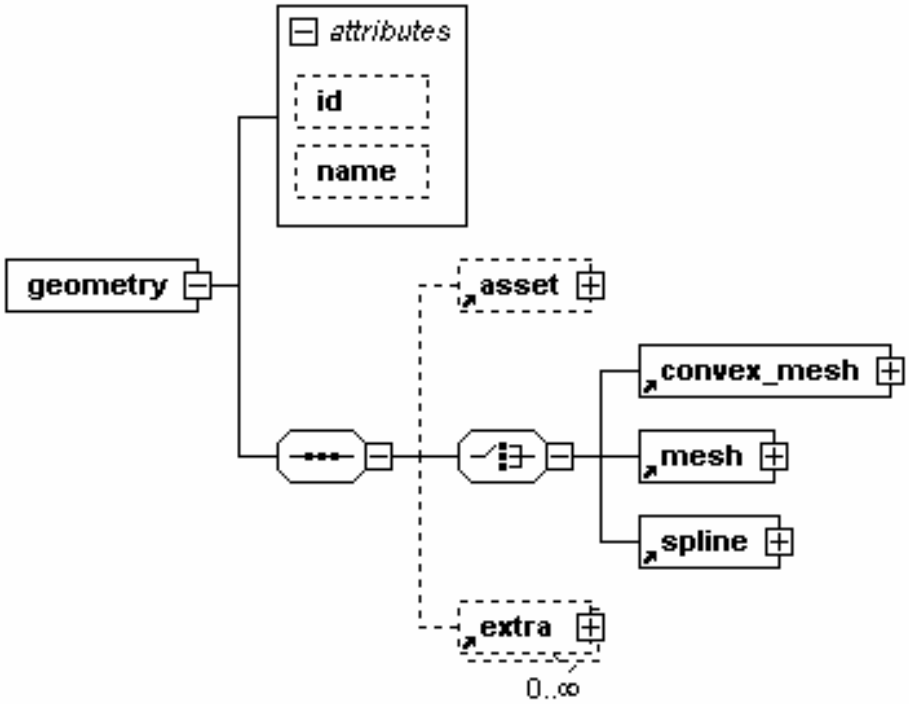
Geometry Schema

- Geometry as data-flow
  - Vertices are still the norm.
    - Attributes are variable.
  - Meshes contain vertices
    - Describes the shape / topology of the geometry.
    - Lots of high frequency data.

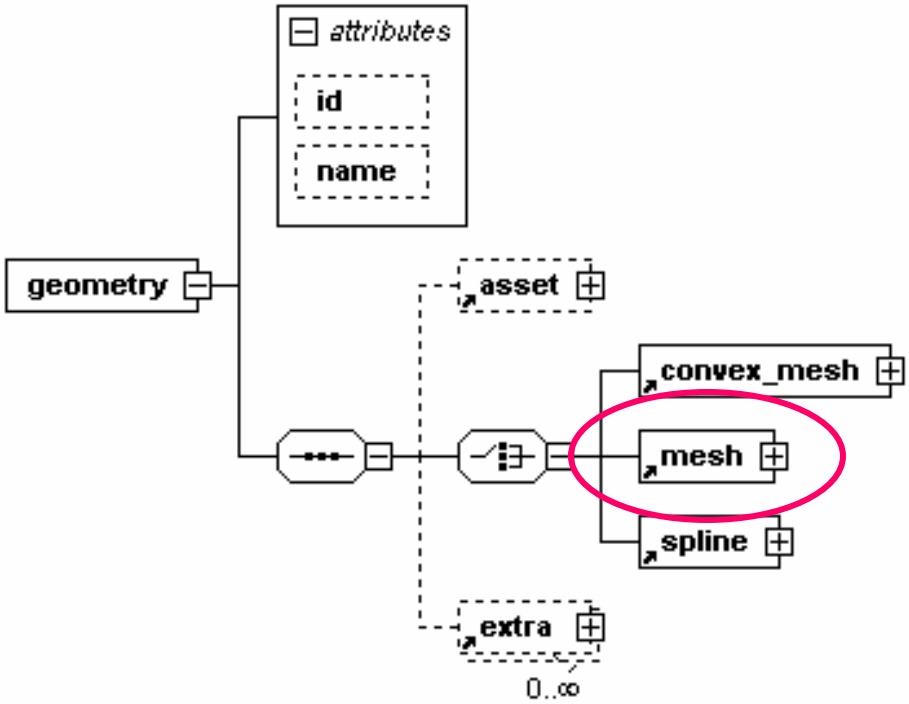
<accessor> (<source>) ➔ <input> (<mesh>)

<geometry>

- Geometry element
  - Contains various types of descriptions.
  - Instantiated into the scene.
  - Stored in the <library_geometies> element

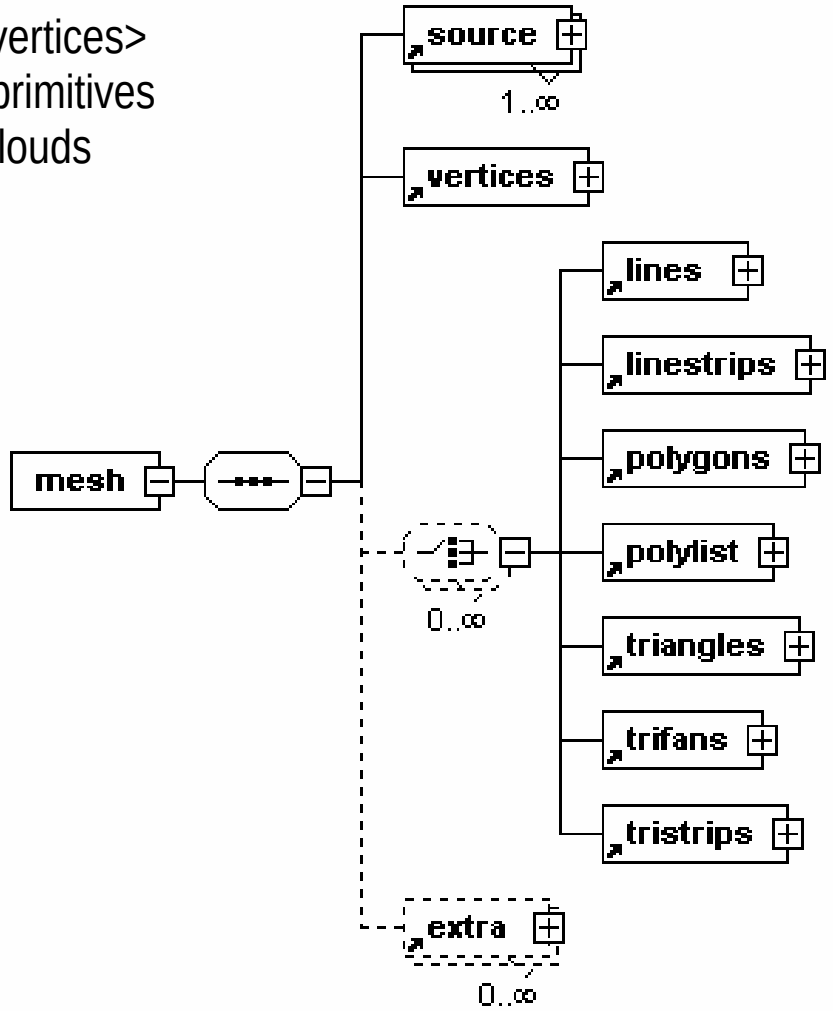


<geometry>

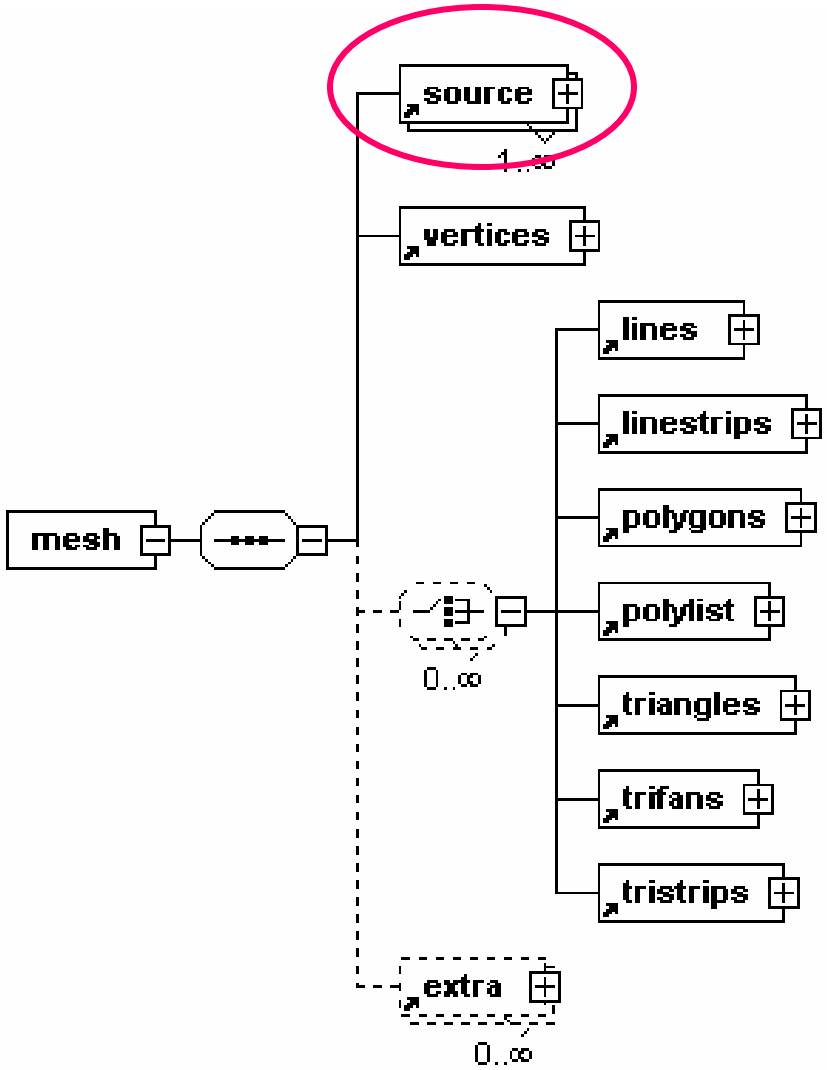


<mesh>

- Mesh element
  - One or more <source>
  - Exactly one <vertices>
  - Zero or more primitives
    - For point clouds

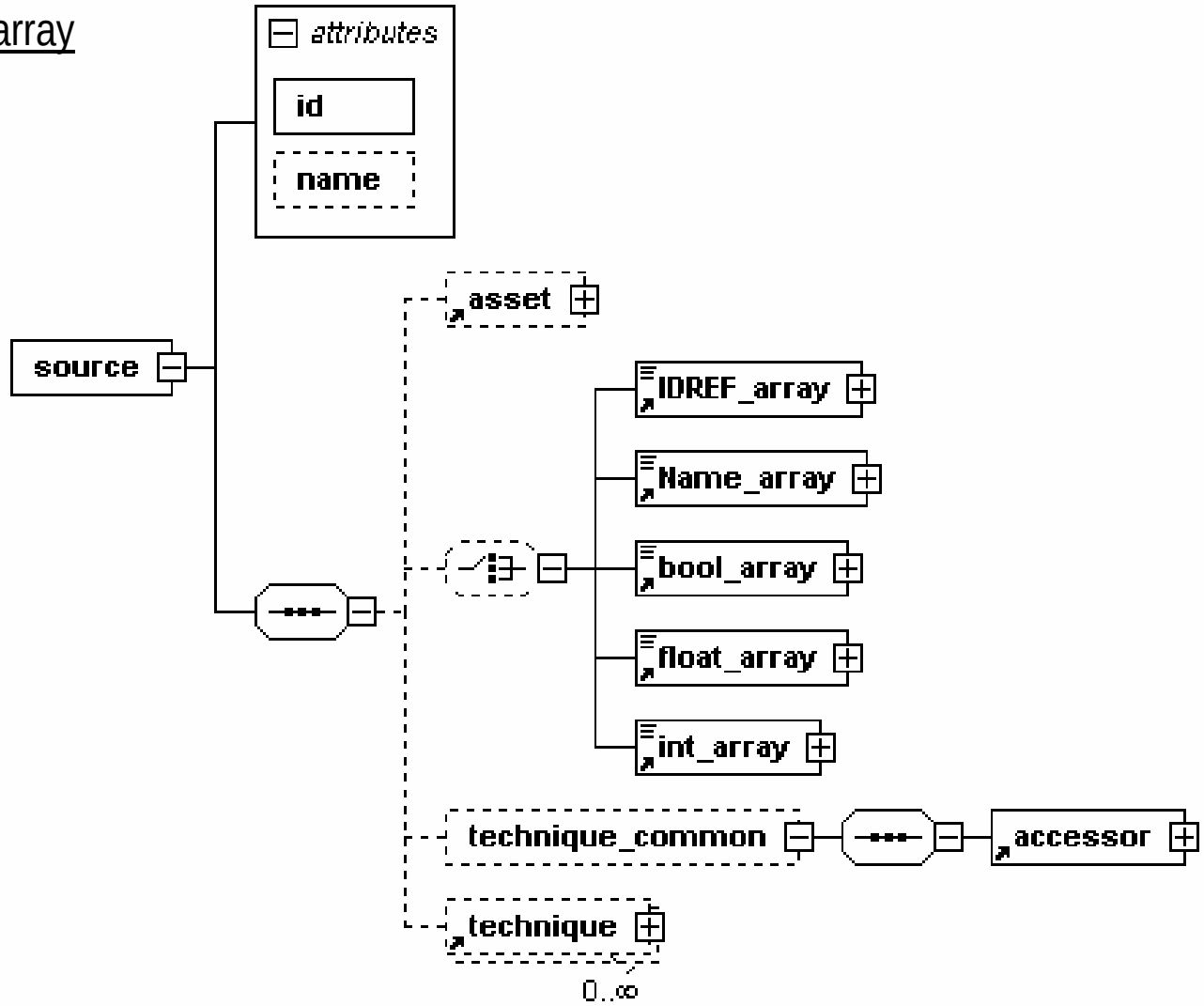


<mesh>



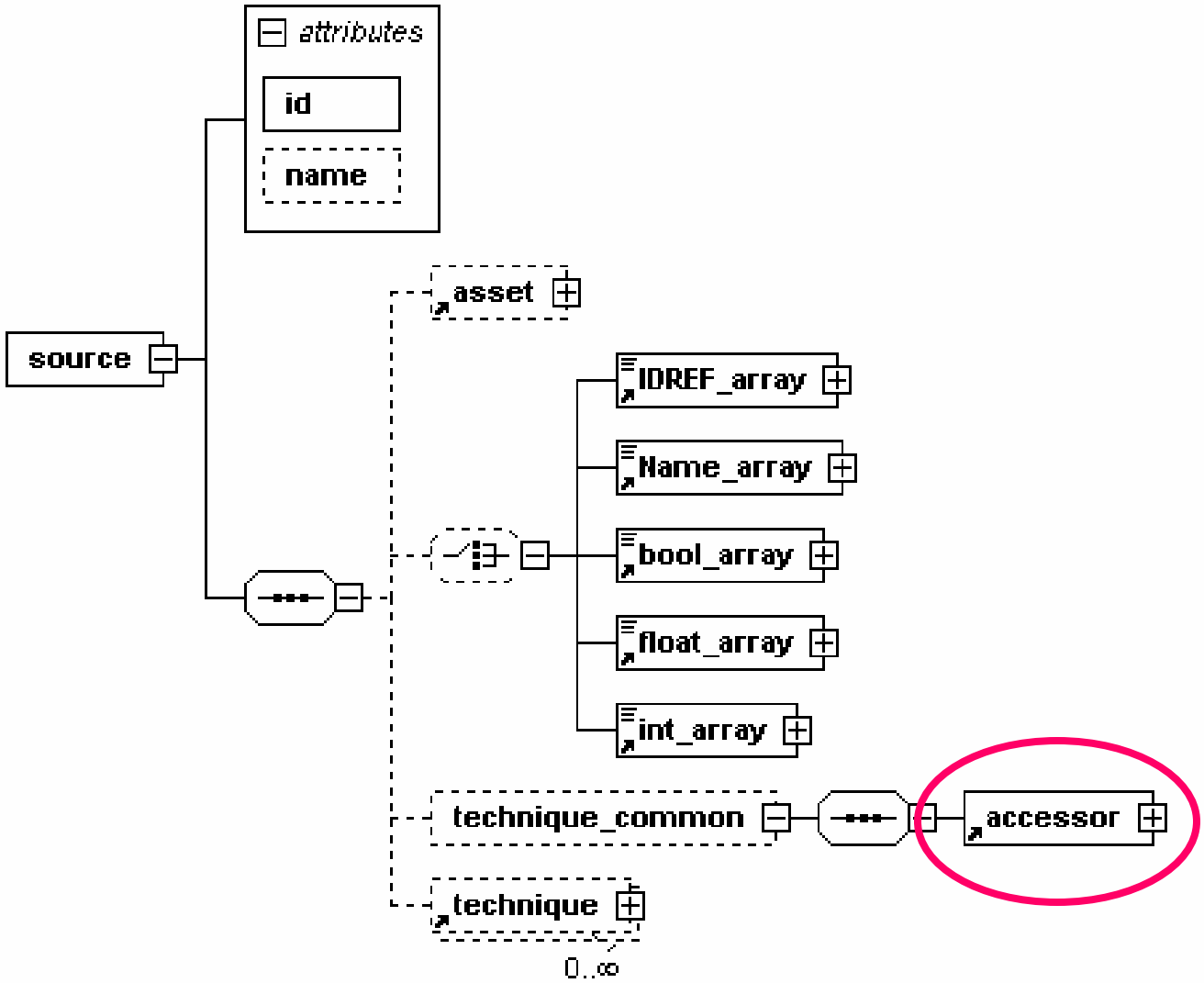
<source>

- Data source element
  - Raw data stream
  - Strongly typed array
  - No semantics



COLLADA

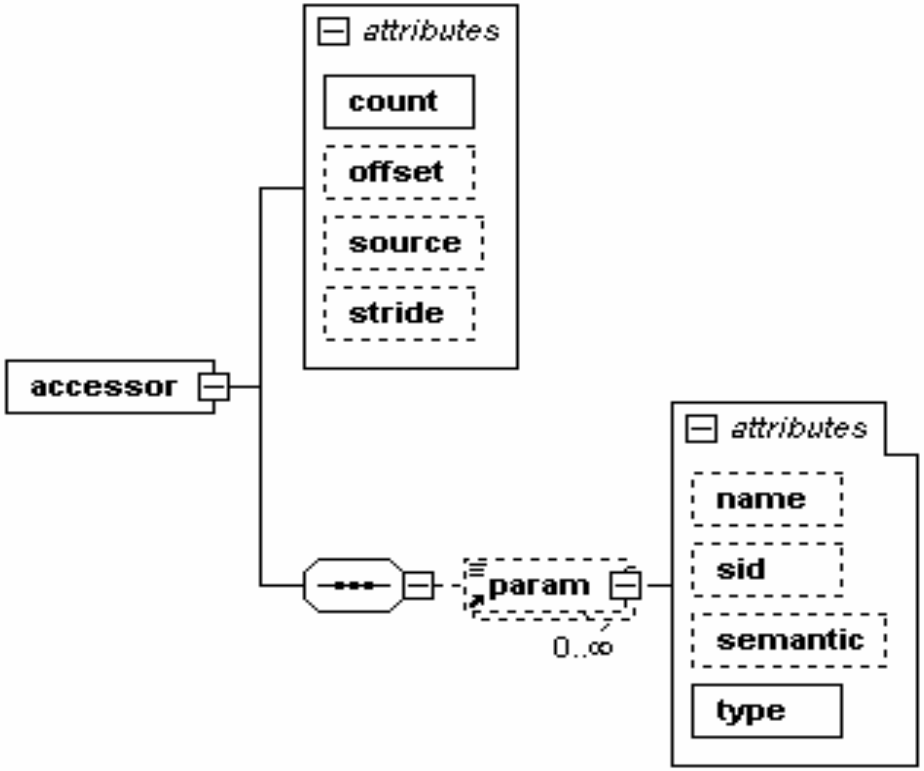
<source>



COLLADA

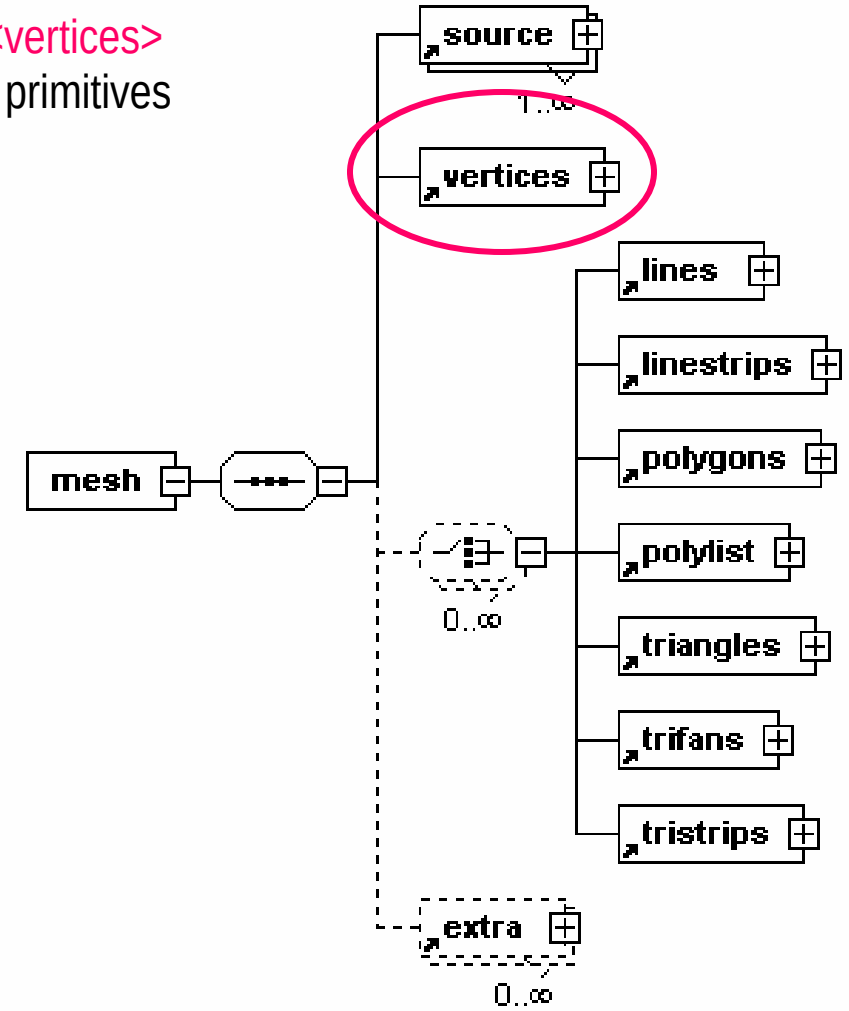
<accessor>

- Data accessor element
  - <source> output interface
  - Describes how to access the source data
  - Data can be stored externally



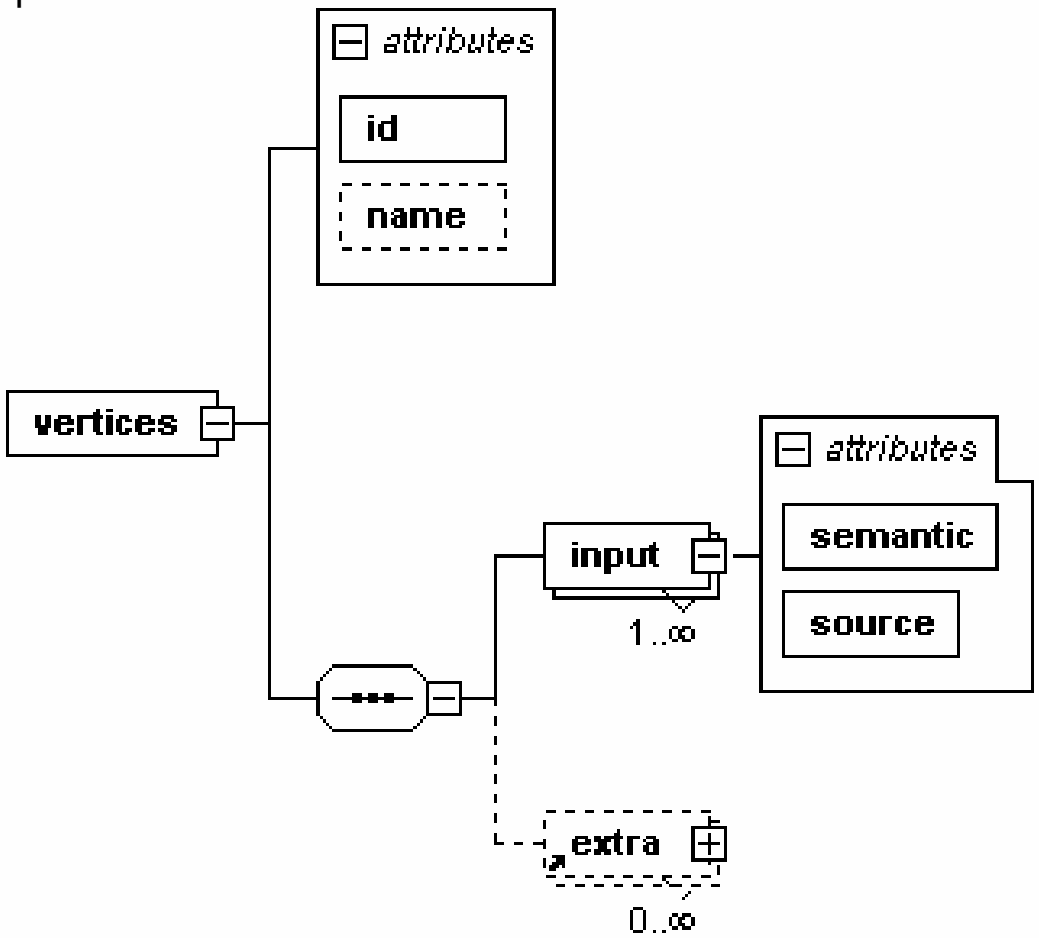
<mesh>

- Mesh element
  - One or more <source>
  - Exactly one <vertices>
  - Zero or more primitives

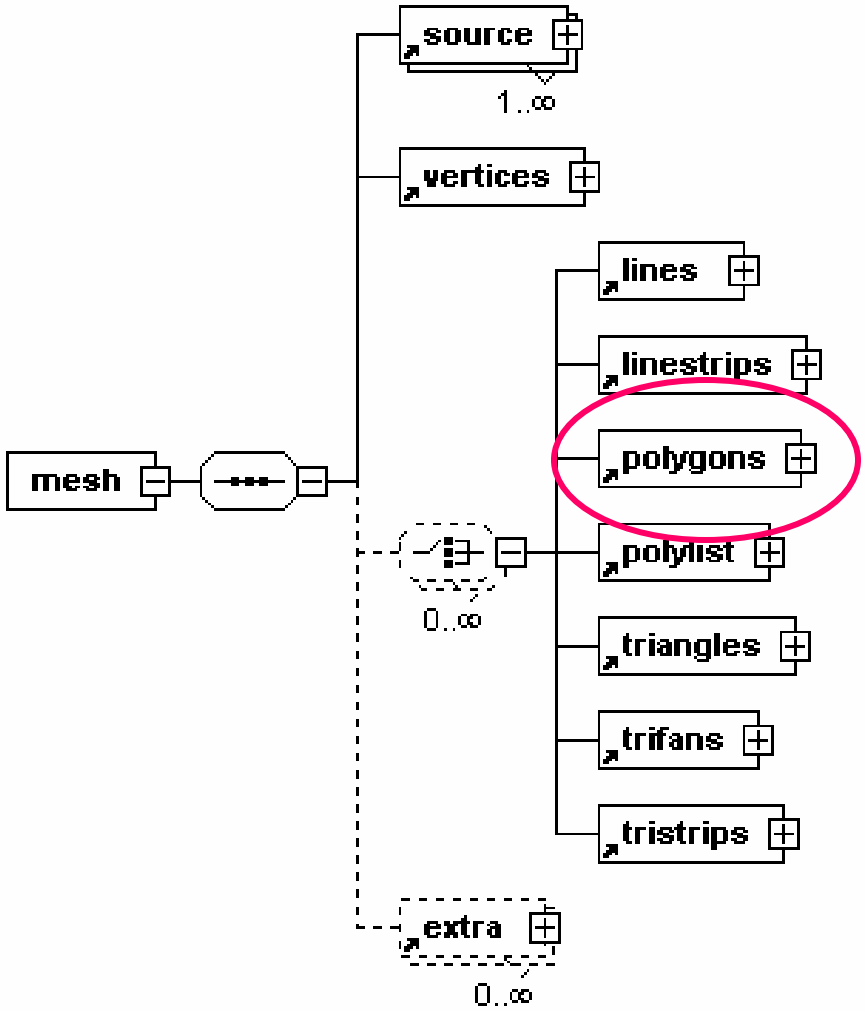


<vertices>

- Mesh vertex element
  - Groups mesh vertex attribute data
  - Collect data independent to primitive assembly
  - “POSITION” semantic required
  - Unlimited attributes
  - nDimentional data OK

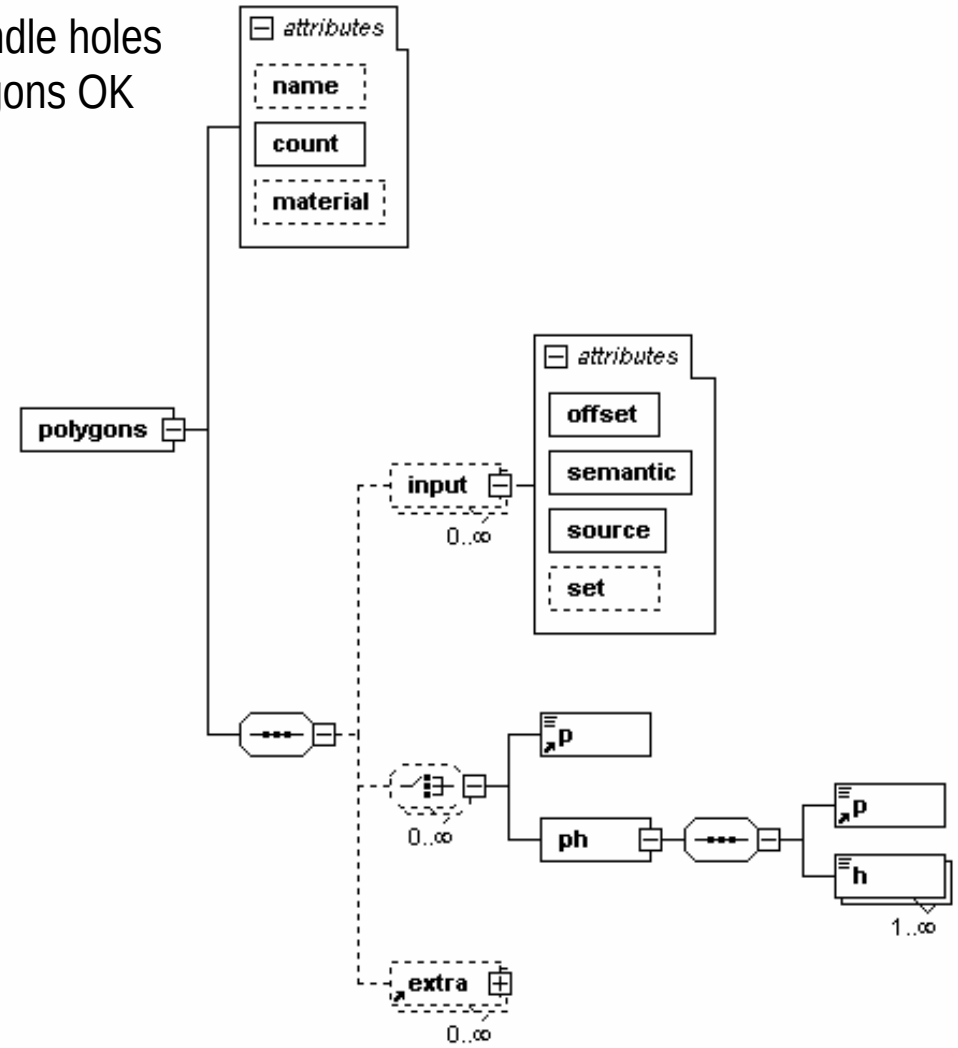


<mesh>



COLLADA <polygons>

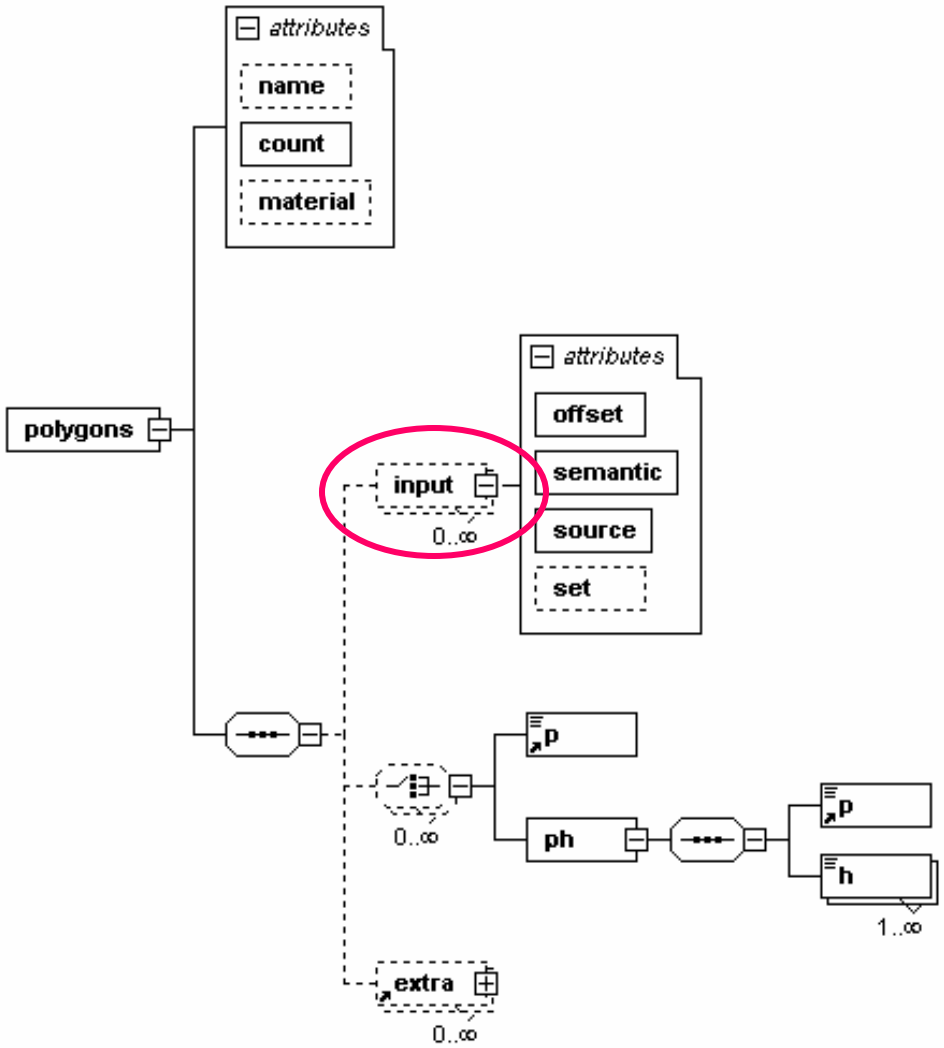
- Polygons element
  - Collates vertices into primitives.
  - General primitive, can handle holes
  - Complex non planar polygons OK



COLLADA <polygons>

- Input streams
  - Possible Input semantics:

VERTEX
TEXCOORD
NORMAL
COLOR
BINORMAL
UV
…



Geometry example (1)

- A super cool box

```
<geometry id="box" name="super_cool_box">
    <mesh>
      <source id="box-Pos">
        <float_array id="box-Pos-array" count="24">
          -0.5  0.5  0.5  0.5  0.5  0.5 -0.5 -0.5  0.5  0.5 -0.5  0.5
          -0.5  0.5 -0.5  0.5  0.5 -0.5 -0.5 -0.5 -0.5  0.5 -0.5 -0.5
        </float_array>
        <technique_common>
         <accessor source="#box-Pos-array" count="8" stride="3">
           <param name="X" type="float"/>
           <param name="Y" type="float"/>
           <param name="Z" type="float"/>
         </accessor>
        </technique_common>
      </source>
      <source id="box-0-Normal">
…..
```

Geometry example (2)

```
…
 <vertices id="box-Vtx">
  <input semantic="POSITION" source="#box-Pos"/>
 </vertices>
 <polygons count="3" material="RED">
  <input semantic="VERTEX" source="#box-Vtx" offset="0"/>
  <input semantic="NORMAL" source="#box-0-Normal" offset="1"/>
  <p>0 4 2 4 3 4 1 4</p> <p>0 2 1 2 5 2 4 2</p> <p>6 3 7 3 3 3 2 3</p>
 </polygons>
 <polygons count="3" material="BLUE">
  <input semantic="VERTEX" source="#box-Vtx" offset="0"/>
  <input semantic="NORMAL" source="#box-0-Normal" offset="1"/>
  <p>0 1 4 1 6 1 2 1</p> <p>3 0 7 0 5 0 1 0</p> <p>5 5 7 5 6 5 4 5</p>
 </polygons>
</mesh>
</geometry>
```

Geometry example (3)

- Vertex data stream

```
<geometry id="box" name="super_cool_box">
    <mesh>
      <source id=  "box-Pos"
        <float_array id="  "box-Pos-array"  count="24">
          ….
        <technique_common>
          <accessor source=  "#box-Pos-array"  count="8" stride="3">
            …
      </source>
      <source id="box-Normal">
        …
      <vertices id=  box-Vtx"
        <input semantic="POSITION" source=  "#box-Pos"
      </vertices>
       <polygons count="3" material="RED">
         <input semantic="VERTEX" source=  "#box-Vtx"  fset="0"/>
          <input semantic="NORMAL" source="#box-Normal" offset="1"/>
…
```

Geometry example (4)

- Vertex and Normal data stream

```
<geometry id="box" name="super_cool_box">
    <mesh>
      <source id="box-Pos">
        <float_array id="box-Pos-array" count="24">
          ….
        <technique_common>
          <accessor source="#box-Pos-array" count="8" stride="3">
            …
      </source>
      <source id="box-Normal">
        …
      <vertices id="box-Vtx">
        <input semantic="POSITION" source="#box-Pos" />
      </vertices>
       <polygons count="3" material="RED">
         <input semantic="VERTEX" source="#box-Vtx" offset="0"/>
         <input semantic="NORMAL" source="#box-Normal" offset="1"/>
…
```

Geometry example (1.3.x - obsolete)

- Can you spot the problems ?

```
<geometry id="box">
    <mesh>
      <source id="box-Position">
        <array name="box-Position-array" type="float" count="24">
          -0.5  0.5  0.5  0.5  0.5  0.5 -0.5 -0.5  0.5  0.5 -0.5  0.5
          -0.5  0.5 -0.5  0.5  0.5 -0.5 -0.5 -0.5 -0.5  0.5 -0.5 -0.5
        </array>
        <technique profile="COMMON1.0">
          <accessor source="#box-Position-array" count="8" stride="3“ />
        </technique>
      </source>
      <polygons count="6" material="#Blue">
        <input semantic="POSITION" source="#box-Position"/>
        <p>0 2 3 1</p> <p>0 1 5 4</p> <p>6 7 3 2</p>
        <p>0 4 6 2</p> <p>3 7 5 1</p> <p>5 7 6 4</p>
      </polygons>
    </mesh>
  </geometry>
```

Geometry example (1.3.x - obsolete)

- Using name for referencing

```
<geometry id="box">
    <mesh>
      <source id="box-Position">
        <array name="box-Position-array" type="float" count="24">
          -0.5  0.5  0.5  0.5  0.5  0.5 -0.5 -0.5  0.5  0.5 -0.5  0.5
          -0.5  0.5 -0.5  0.5  0.5 -0.5 -0.5 -0.5 -0.5  0.5 -0.5 -0.5
        </array>
        <technique profile="COMMON1.0">
          <accessor source="#box-Position-array" count="8" stride="3“ />
        </technique>
      </source>
      <polygons count="6" material="#Blue">
        <input semantic="POSITION" source="#box-Position"/>
        <p>0 2 3 1</p> <p>0 1 5 4</p> <p>6 7 3 2</p>
        <p>0 4 6 2</p> <p>3 7 5 1</p> <p>5 7 6 4</p>
      </polygons>
    </mesh>
  </geometry>
```

Geometry example (1.3.x - obsolete)

- Weakly typed elements

```
<geometry id="box">
    <mesh>
      <source id="box-Position">
        <array name="box-Position-array" type="float" count="24">
          -0.5  0.5  0.5  0.5  0.5  0.5 -0.5 -0.5  0.5  0.5 -0.5  0.5
          -0.5  0.5 -0.5  0.5  0.5 -0.5 -0.5 -0.5 -0.5  0.5 -0.5 -0.5
        </array>
        <technique profile="COMMON1.0">
          <accessor source="#box-Position-array" count="8" stride="3“ />
        </technique>
      </source>
      <polygons count="6" material="#Blue">
        <input semantic="POSITION" source="#box-Position"/>
        <p>0 2 3 1</p> <p>0 1 5 4</p> <p>6 7 3 2</p>
        <p>0 4 6 2</p> <p>3 7 5 1</p> <p>5 7 6 4</p>
      </polygons>
    </mesh>
  </geometry>
```

Geometry example (1.3.x - obsolete)

- Lacking mesh vertex identification

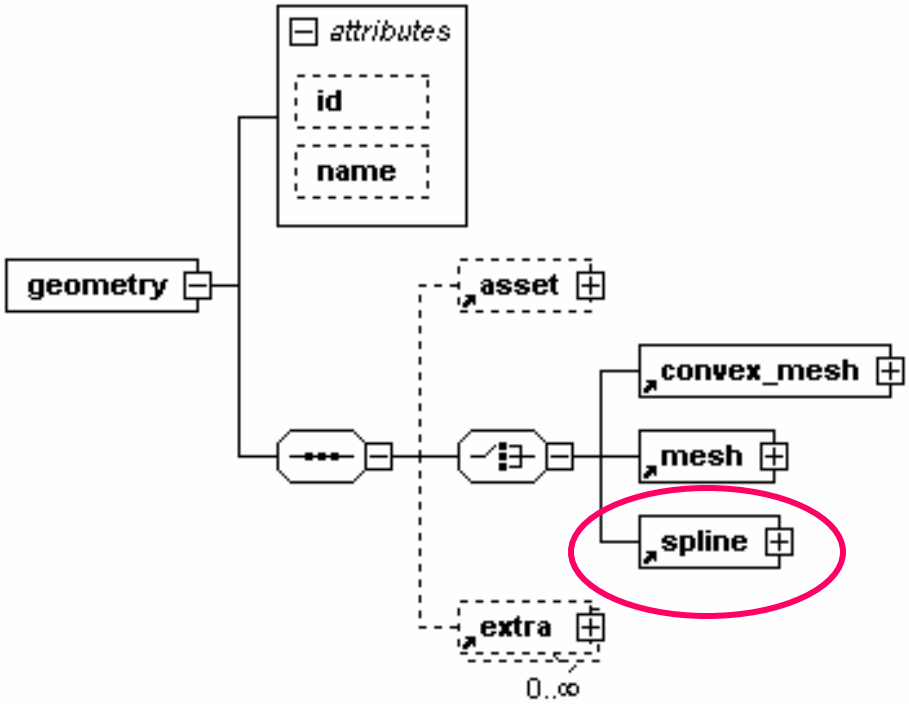
```
<geometry id="box">
    <mesh>
      <source id="box-Position">
        <array name="box-Position-array" type="float" count="24">
          -0.5  0.5  0.5  0.5  0.5  0.5 -0.5 -0.5  0.5  0.5 -0.5  0.5
          -0.5  0.5 -0.5  0.5  0.5 -0.5 -0.5 -0.5 -0.5  0.5 -0.5 -0.5
        </array>
        <technique profile="COMMON1.0">
          <accessor source="#box-Position-array" count="8" stride="3“ />
        </technique>
      </source>
      <vertices id="box-Vtx">
      <input semantic="POSITION" source="#box-Pos"/>
      </vertices>
      <polygons count="6" material="#Blue">
        <input semantic="POSITION" source="#box-Position"/>
        <p> ……
```

Geometry example (1.3.x - obsolete)

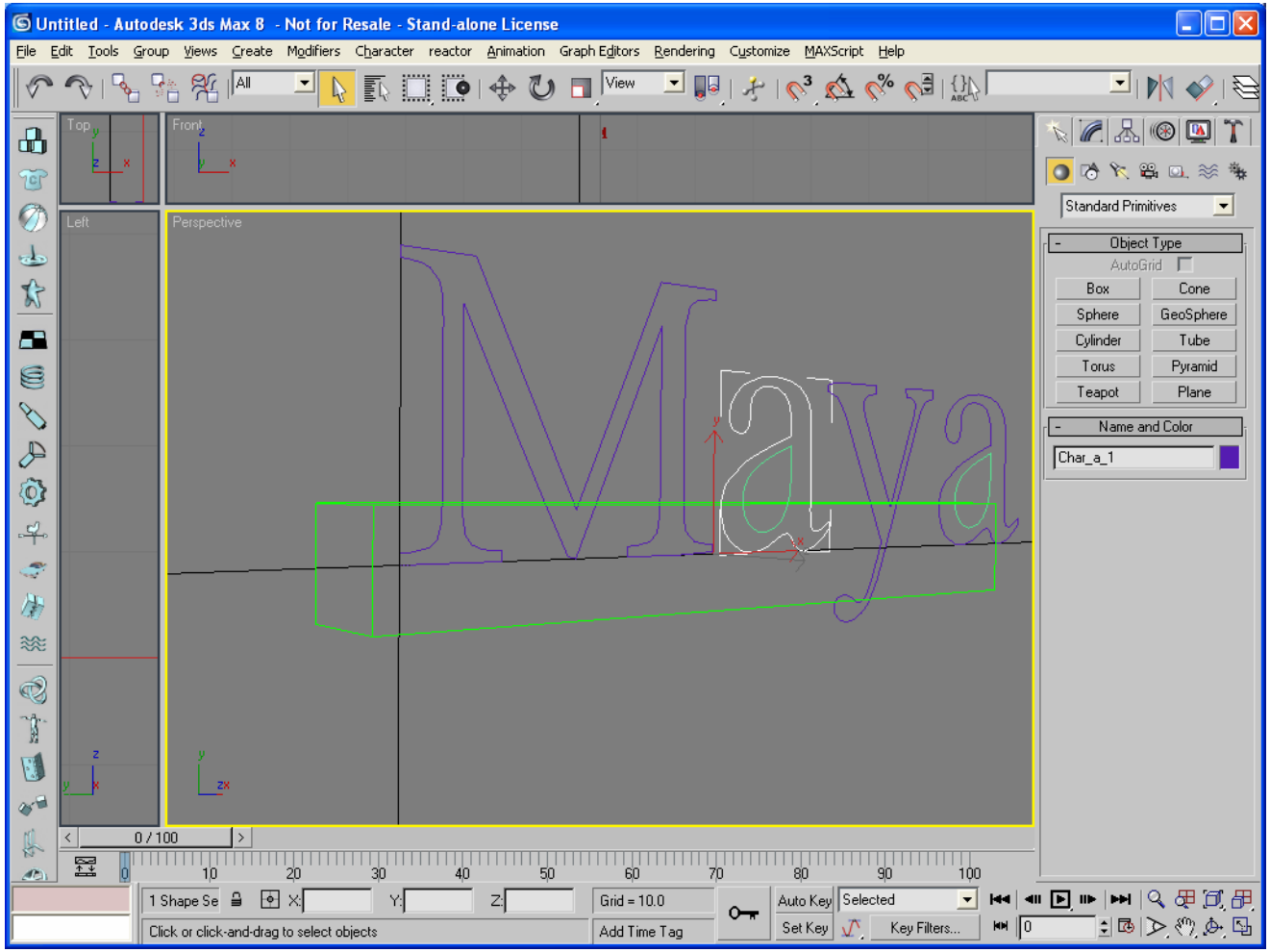
- No ability to instance with different materials

```
<geometry id="box">
    <mesh>
      <source id="box-Position">
        <array name="box-Position-array" type="float" count="24">
          -0.5  0.5  0.5  0.5  0.5  0.5 -0.5 -0.5  0.5  0.5 -0.5  0.5
          -0.5  0.5 -0.5  0.5  0.5 -0.5 -0.5 -0.5 -0.5  0.5 -0.5 -0.5
        </array>
        <technique profile="COMMON1.0">
          <accessor source="#box-Position-array" count="8" stride="3“ />
        </technique>
      </source>
      <polygons count="6" material="#Blue">
        <input semantic="POSITION" source="#box-Position"/>
        <p>0 2 3 1</p> <p>0 1 5 4</p> <p>6 7 3 2</p>
        <p>0 4 6 2</p> <p>3 7 5 1</p> <p>5 7 6 4</p>
      </polygons>
    </mesh>
  </geometry>
```

<geometry>

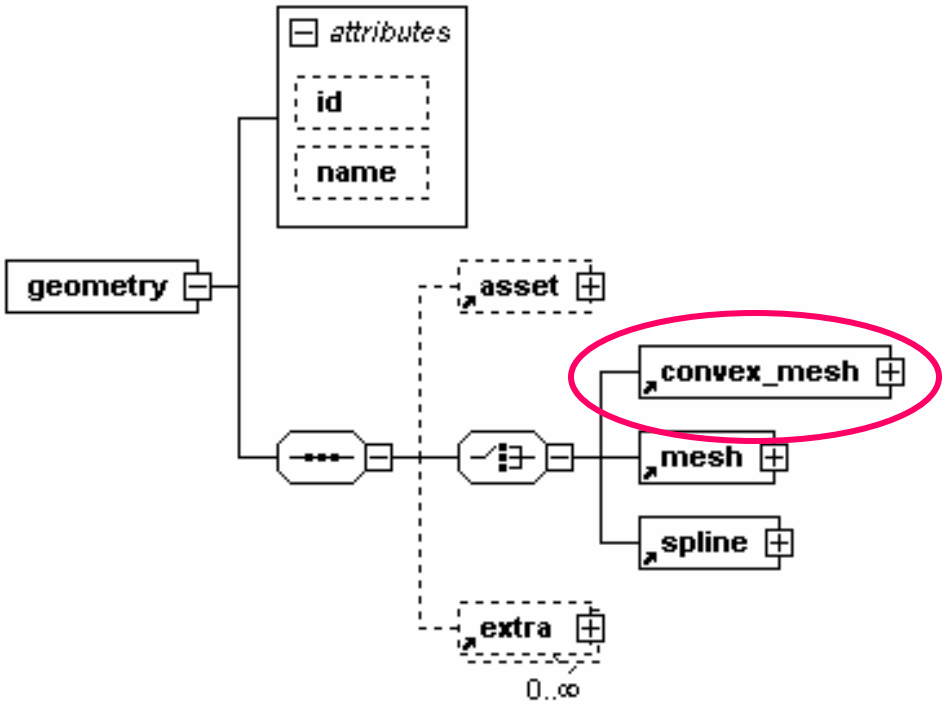


ColladaMax Screenshots

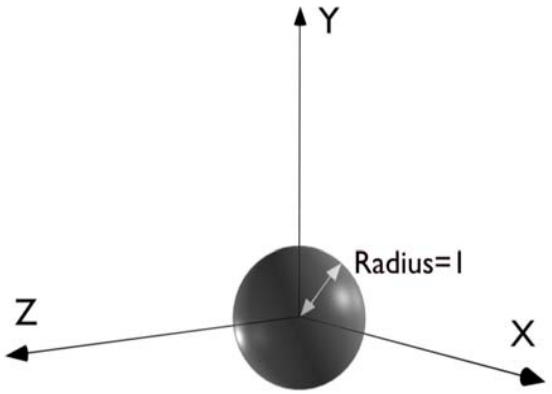


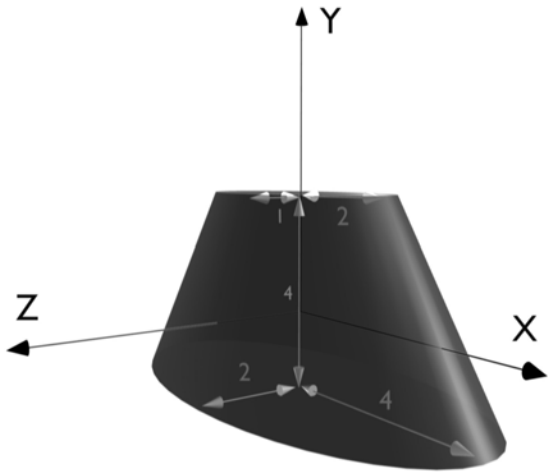
NURBS curves imported from Maya

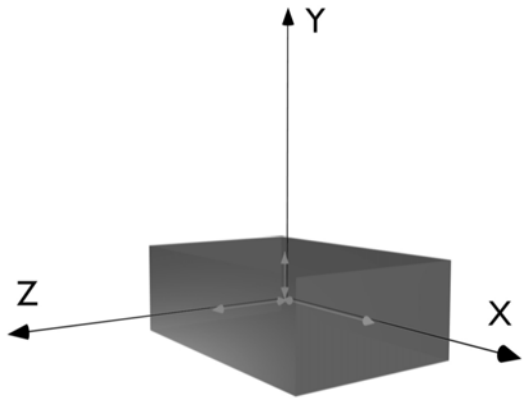
<geometry>

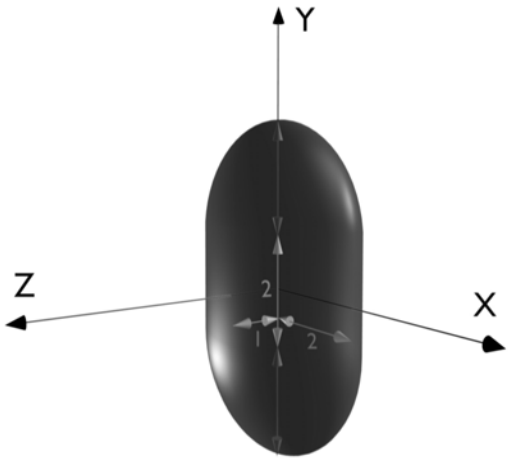


Analytic shapes









Material

Material

- Assign material to geometry symbolically

```
…
 <vertices id="box-Vtx">
  <input semantic="POSITION" source="#box-Pos"/>
 </vertices>
 <polygons count="3" material="RED">
  <input semantic="VERTEX" source="#box-Vtx" offset="0"/>
  <input semantic="NORMAL" source="#box-0-Normal" offset="1"/>
  <p>0 4 2 4 3 4 1 4</p> <p>0 2 1 2 5 2 4 2</p> <p>6 3 7 3 3 3 2 3</p>
 </polygons>
 <polygons count="3" material="BLUE">
  <input semantic="VERTEX" source="#box-Vtx" offset="0"/>
  <input semantic="NORMAL" source="#box-0-Normal" offset="1"/>
  <p>0 1 4 1 6 1 2 1</p> <p>3 0 7 0 5 0 1 0</p> <p>5 5 7 5 6 5 4 5</p>
 </polygons>
</mesh>
</geometry>
```

Instance and material binding

- <library_visual_scenes>
  - Document can store many scenes
- <scene>
  - Default scene for this document
  - <instance_visual_scene url="#...
- <node name="Box" id="Box">
  - Scene contains nodes
- <instance_geometry url="box">
  - Instance geometries from <library_geometries>
- <bind_material>
  - This is how instanced geometry get their final material
- <technique_common>
  <instance_material symbol="RED" target="#redMaterial"/>
  <instance_material symbol="BLUE" target="#blueMaterial"/>
  - Now RED and BLUE have a real material bound

What is a <material>, an <effect>

- Stored in the <library_materials>
- <material> is an instanced of an <effect>
- <effect> is stored in <library_effects>
- A document containing effects is called a COLLADA FX document
- In short an effect is a full list of commands for the GPU subsystem
- Since there is no standard, COLLADA FX is using profiles:
- CG profile (cross API/platform), GL 2.0 (GLSL), DX (HLSL), OpenGL ES 1.x, OpenGL ES 2.0 (GLSL ES)
- <profile_COMMON> effect defines some basic rendering for prototyping and basic interchange
  - *Constant, Lambert, Phong, Blinn*
  - Mandatory for conformant files, can have several profiles for same effect

Texture Mapping (1)

- <image>

```
…
<image id="img01">
   <init_from> image_file.tga</init_from>
```

Texture Mapping (2)

- <surface>

```
<image id="img01">
  <init_from> image_file.tga</init_from>
...
<effect>
…
 <newparam sid="surface">
    <surface> <init_from>img01</init_from> </surface>
 </newparam>
```

Texture Mapping (3)

- <sampler>

```
<image id="img01">
   <init_from> image_file.tga</init_from>
... <effect> …
 <newparam sid="surface">
      <surface> <init_from>img01</init_from> </surface>
</newparam>
 <newparam sid="sampler">
   <sampler2D> <source>surface</source> </sampler2D>
 </newparam>
```

Texture Mapping (4)

- <texture>

```
<image id="img01">
   <init_from> image_file.tga</init_from>
... <effect> …
 <newparam sid="surface">
      <surface> <init_from>img01</init_from> </surface>
</newparam>
 <newparam sid="sampler">
   <sampler2D> <source>surface</source> </sampler2D>
 </newparam>
 <profile_COMMON> <phong>
    <diffuse> <texture texture="sampler" texcoord=“myUVs"/>
```

Texture Mapping (5)

- <material>

```
<image id="img01">
   <init_from> image_file.tga</init_from>
... <effect> …
 <newparam sid="surface">
      <surface> <init_from>img01</init_from> </surface>
</newparam>
 <newparam sid="sampler">
   <sampler2D> <source>surface</source> </sampler2D>
 </newparam>
 <profile_COMMON> <phong>
    <diffuse> <texture texture="sampler" texcoord="myUVs"/>
… <material> …
<material id="material_id">
 <instance_effect url="#effect_id" />
</material>
```

Texture Mapping (6)

- <material> can change <effect> parameter when instancing

```
<image id="img01">
  <init_from> image_file.tga</init_from>
... <effect> …
 <newparam sid="surface">
     <surface> <init_from>img01</init_from> </surface>
</newparam>
 <newparam sid="sampler">
  <sampler2D> <source>surface</source> </sampler2D>
 </newparam>
 <profile_COMMON> <phong>
   <diffuse> <texture texture="sampler" texcoord=“myUVs"/>
… <material> …
<material id="material_id">
 <instance_effect url="#effect_id" >
  <setparam ref=“surface”> <surface>
    <init_from> img02 </init_from></surface>
 </instance_effect>
</material>
```

Texture Mapping (7)

- <geometry> has to have “TEXCOORD” *input* in addition to *material*

```
...<geometry>…
<geometry id=“geometry”>
  <input semantic=“TEXCOORD” source=“#...” set=“123”
  <triangles material=“material_symbol” …
</geometry>
```

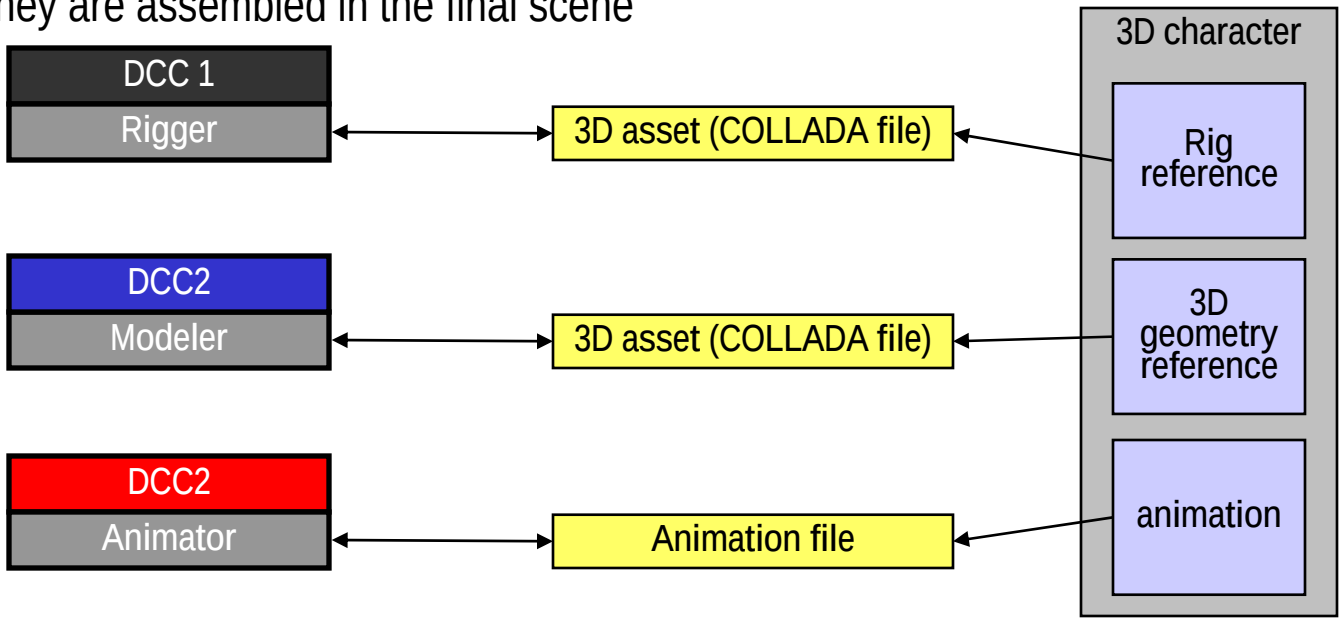
Texture Mapping (8)

- Binding can now happen when instancing the geometry

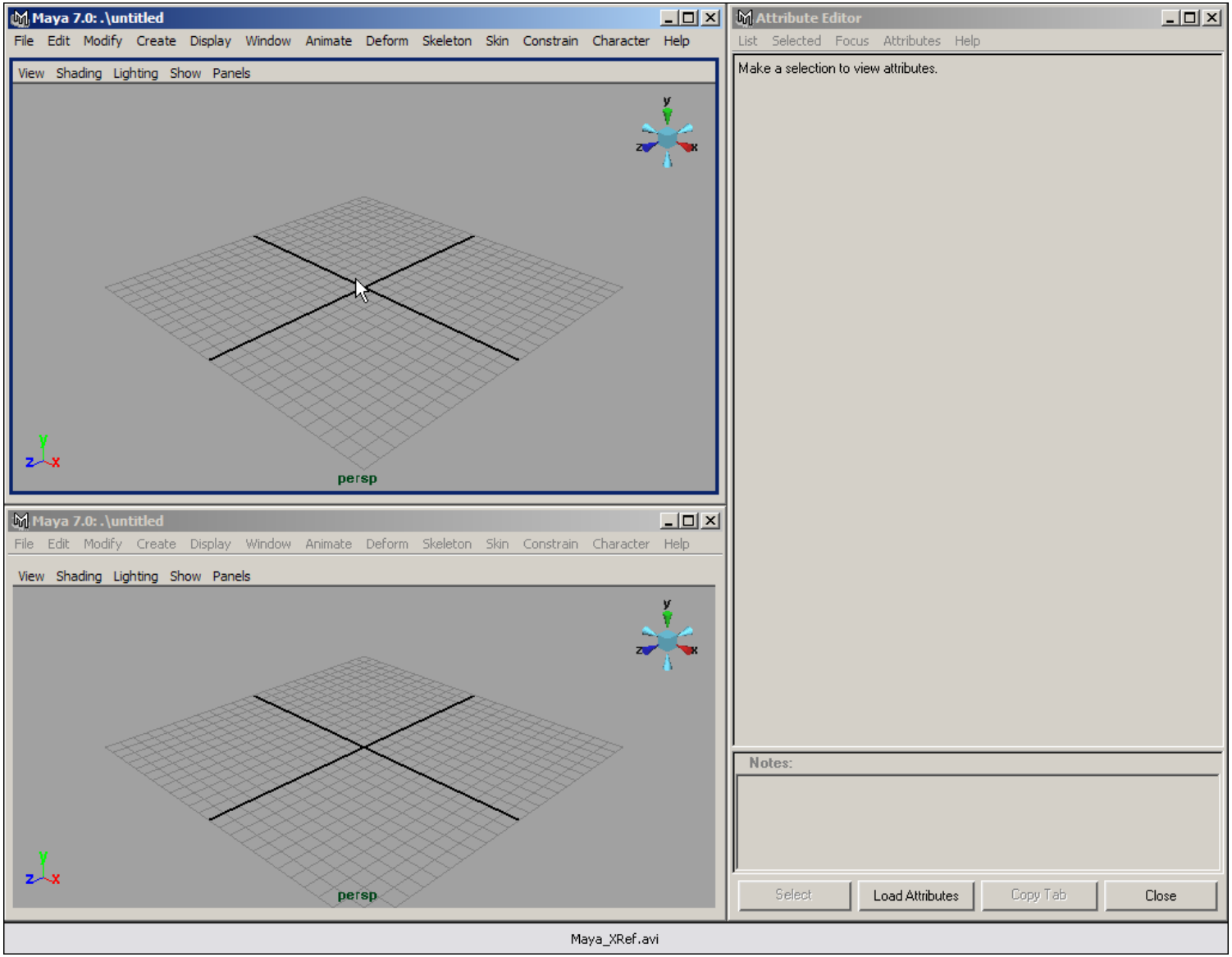
```
...<geometry>…
<geometry id=“geometry”>
  <input semantic=“TEXCOORD” source=“#...” set=“123”
  <triangles material=“material_symbol” …
</geometry>

… <scene>…
<instance_geometry>
  <bind_material>
    <instance_material symbol="material_symbol" target="#material_id">
      <bind_vertex_input semantic="myUVs"
     input_semantic="TEXCOORD" set=“123” />
     <bind_vertex_input semantic=“otherUVs“
     ….
```

COLLADA in workgroups and collaborative workspaces

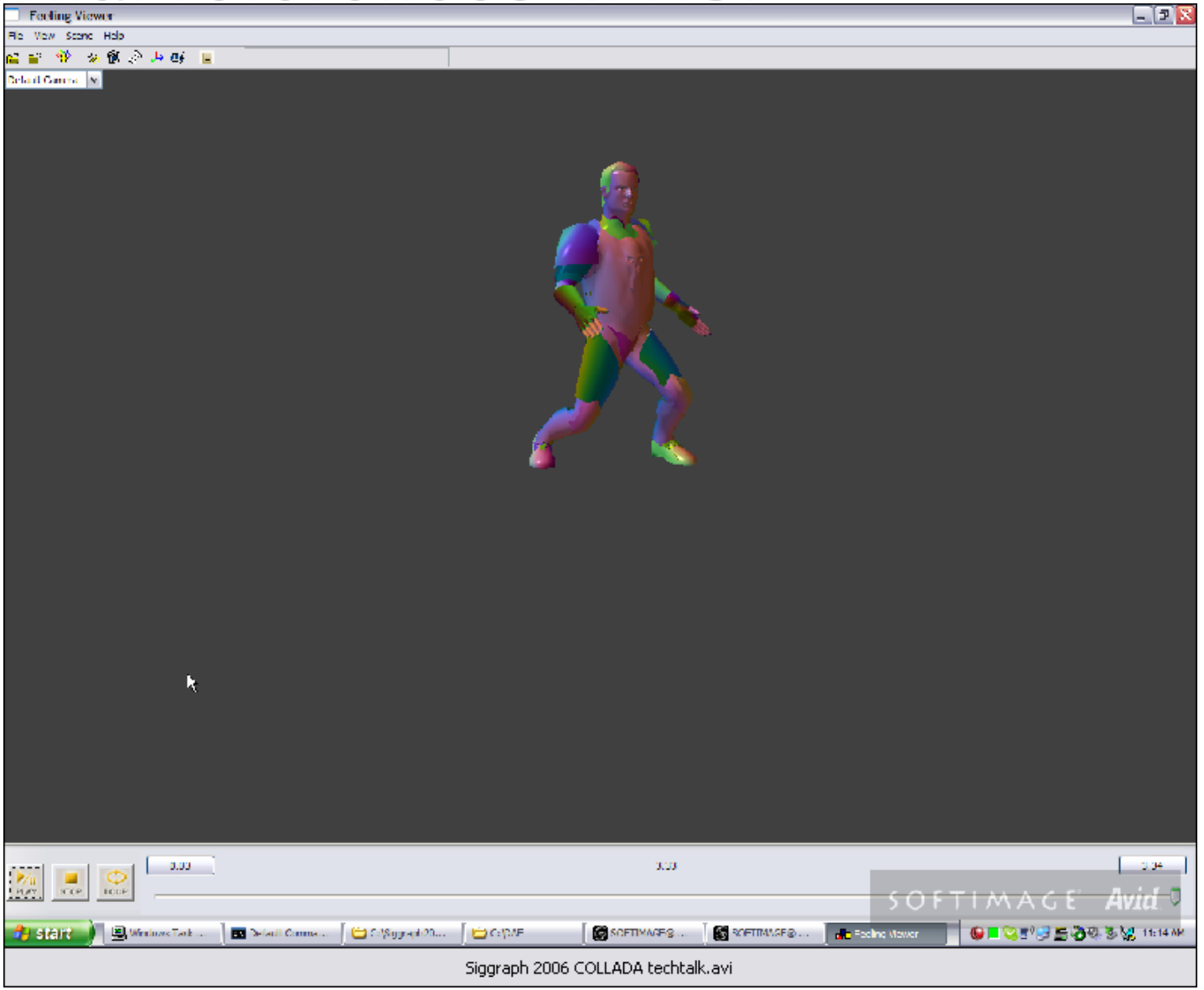
- Taking advantage of the URI technology
- Assembly of scene involving asset created by multiple parties
  - Use reference models to externally reference 3D assets
- Multi task on 3D assets (i.e. rigging, modeling, animating)
  - Rig, geometry and animation are stored in different COLLADA files
  - They are assembled in the final scene



External references using Maya



External references in XSI



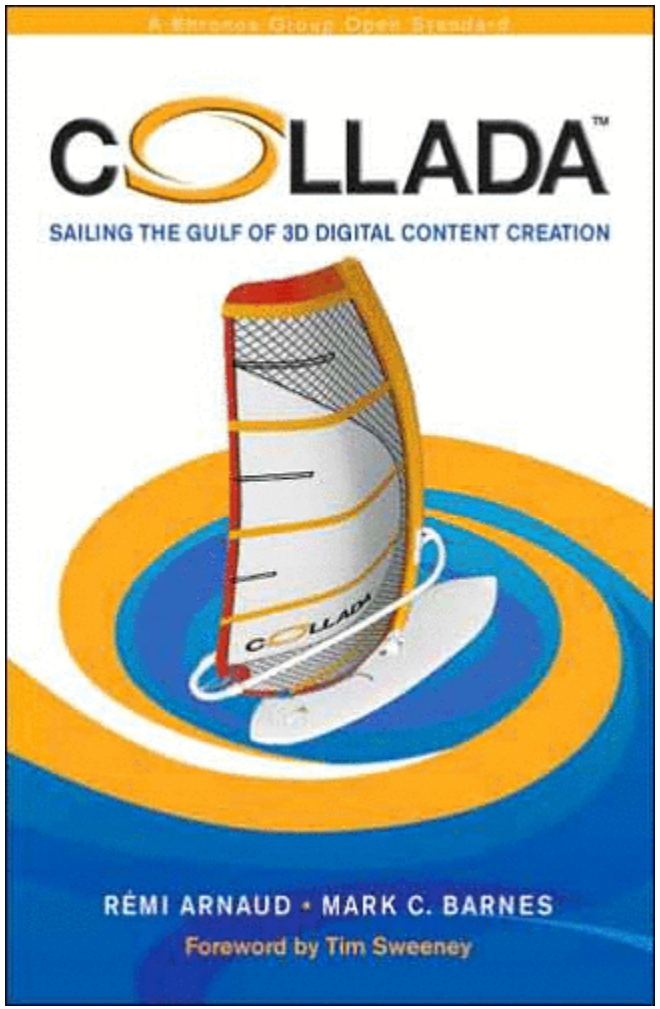
DOCUMENTATION

Lots of documentation

- COLLADA Schema is the formal definition – khronos.org/collada
  - Tools can help humans to read it – XMLSpy (free home edition)
    - Schema is XML, no need to learn another language
  - Auto-generate the COLLADA DOM front end API and documentation
  - Leverage XML Schema language as much as we can
    - Design schema with this in mind (strongly typed …)
- COLLADA Specification – khronos.org/collada
  - pdf format, available in English and Japanese
  - 298 pages
  - Tools requirements
  - Additional rules not encoded in the Schema
    - Count element has to be equal to the number of values in an array
- Collada.org
  - Forum for discussions and Q&A
- Feelingsoftware.org
  - Online docunentation for Max/Maya/FCOLLADA/Viewer

and fresh from the press....

COLLADA Book



Hardcover ~250 pages

Guide to 1.4.x specification

Idea came from Eurographics'05

Provides insight to why COLLADA is designed this way

Gives hints to how COLLADA may evolve

Provides a bit of history

Publisher: AK Peters

Book signing Thursday 10h30

TOOLS

Unreal Engine 3 COLLADA import

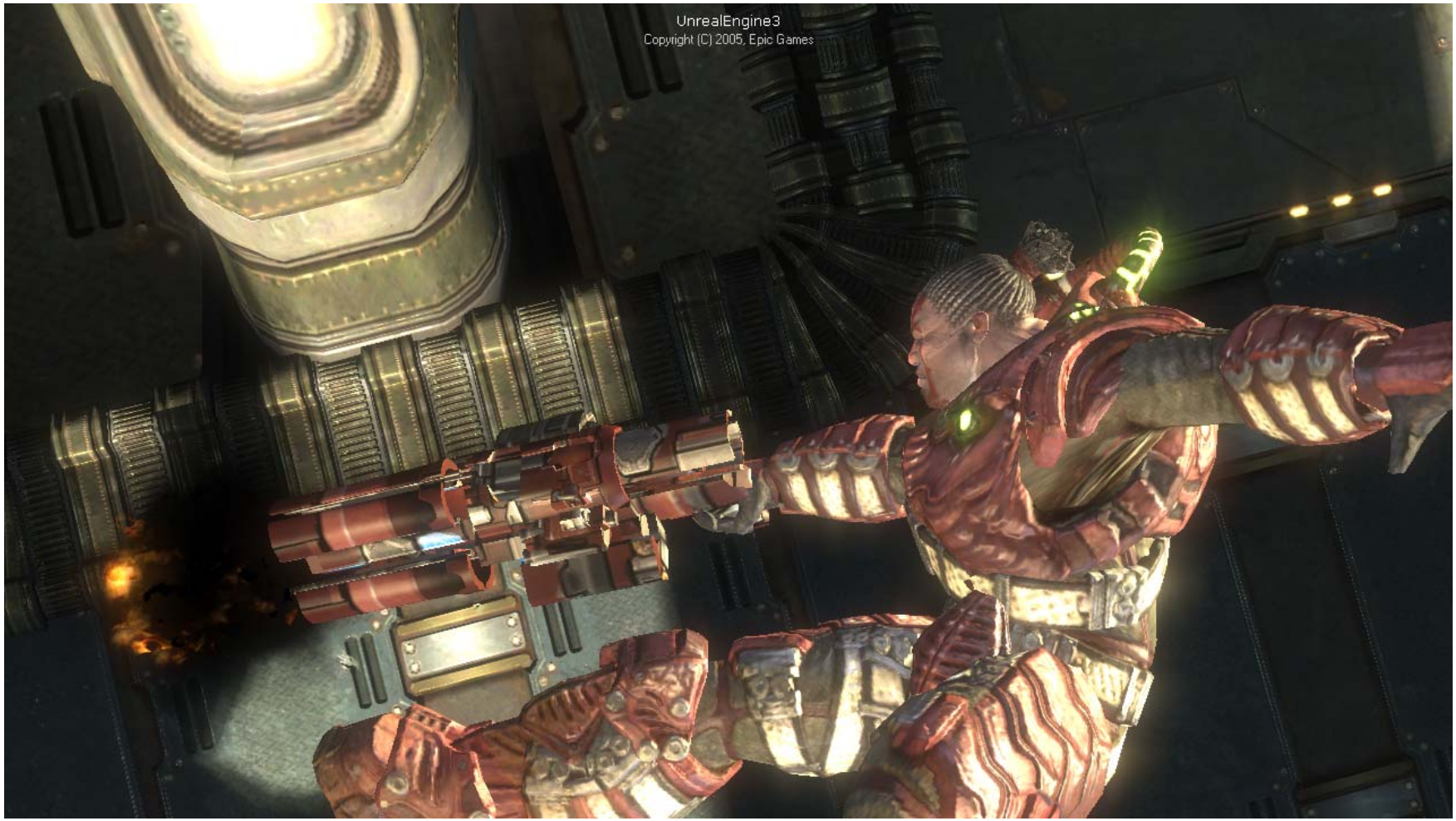


Image courtesy of Epic Games (www.unrealtechnology.com)

Penumbra

Image courtesy of Frictional Games (www.frictionalgames.com)

Game engines

Unreal 3D - *not* free, source license available

Ogre – Open source, free

C4 Engine (cheap)

Irrlicht Engine – Open source, free

PS3 SDK – COLLADA incorporated

…

Duck in Foster City – Google Earth

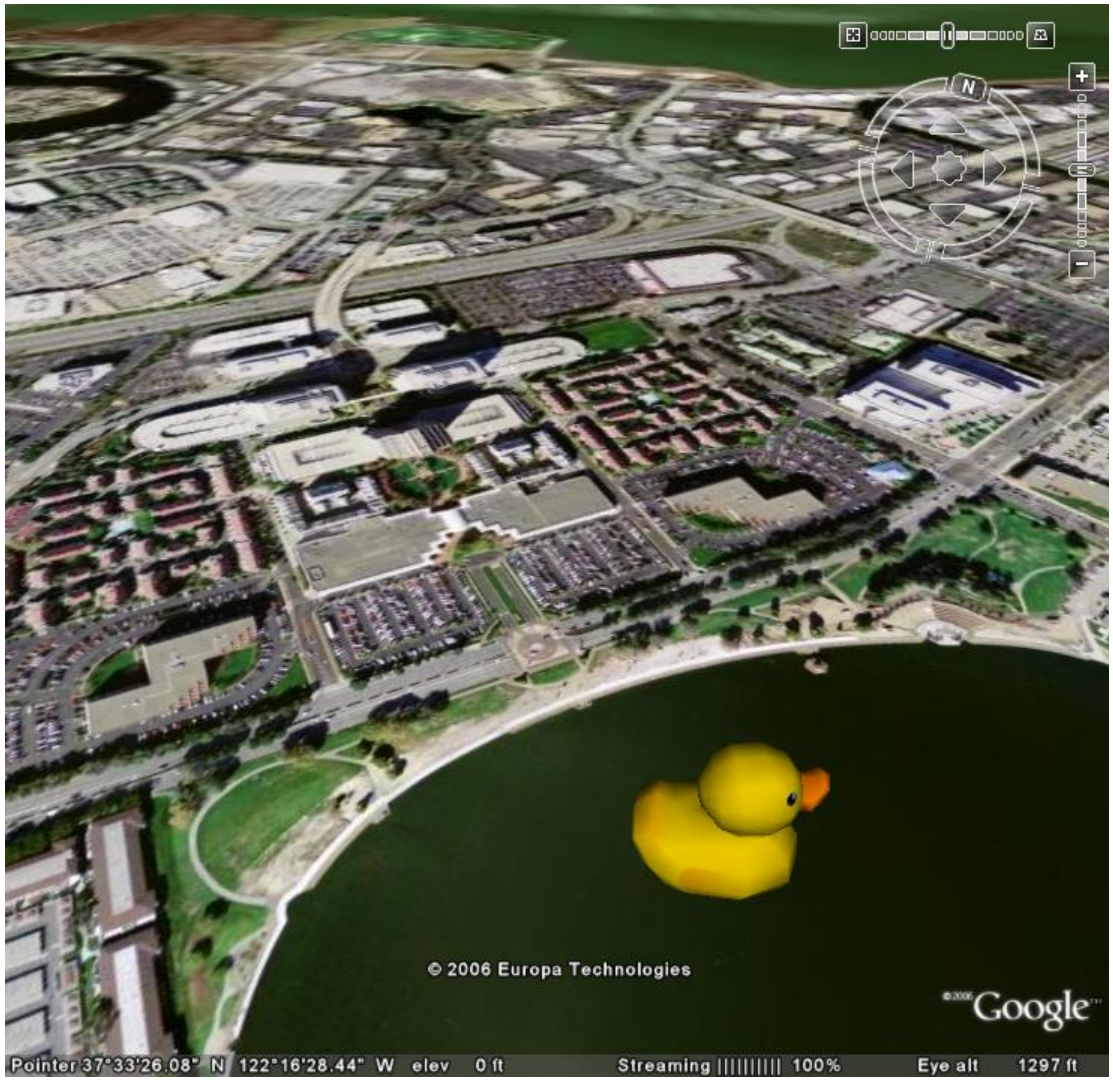


Image courtesy of Google and SCE (earth.google.com)

COLLADA duck in Google Earth ripples

- SketchUp COLLADA export (kmz)
- Real Vis, PhotoModeler, Autocad suite, …
- Many other business oportunities

- Free modeler, free earth viewer, and free content !
  - www.warehouse3d.com
  - Current implementation in Beta
  - KMZ fixer conditioner

Warehouse 3d

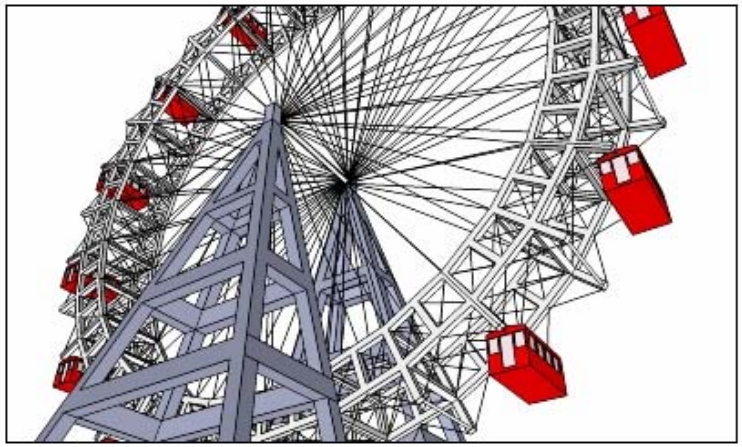
Google 3D Warehouse

Search | Get the 3D Warehouse Network Link

e.g. "airplane", "taj mahal", "trees"

Model of Wiener Riesenrad - Ferris Wheel Vienna by denyo1980

Uploaded on May 1, 2006



Report a policy violation

Download to Google SketchUp | View in Google Earth

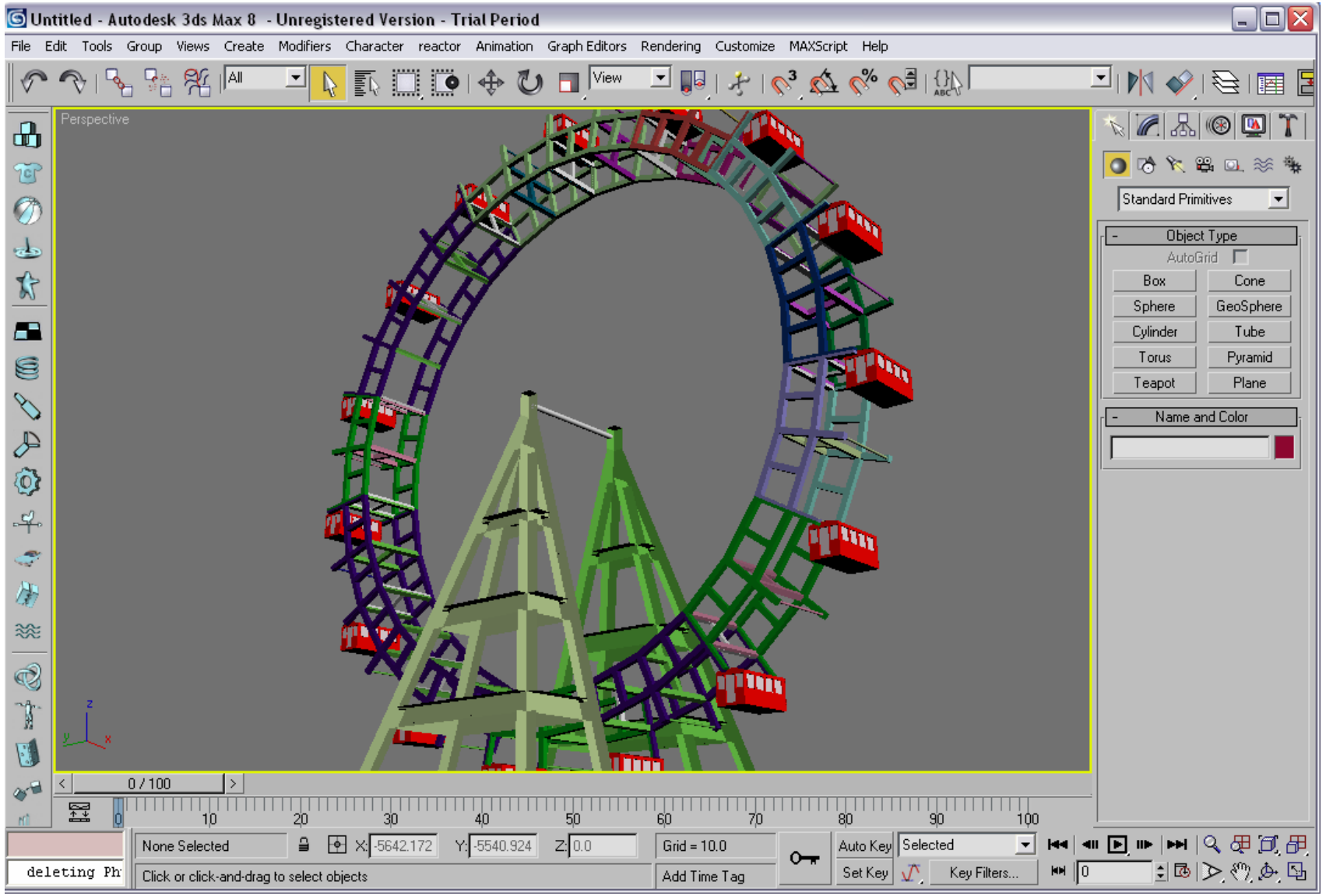
Tags

Wien, Vienna, ferris wheel, ferris, wheel, austria, österreich, wiener riesenrad, wiener, riesenrad, landmark, capital, memorial, famous, building

Description

it was the highest ferris wheel in the world until the millenium wheel in london was built. but this one was built 1896!!
http://www.wienerriesenrad.com/cgi-bin/tagnacht.cgi?sprache=englisch&site=home.htm

A minute later



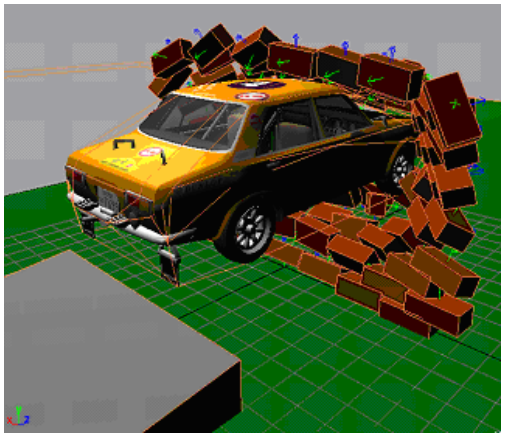
DCC tools

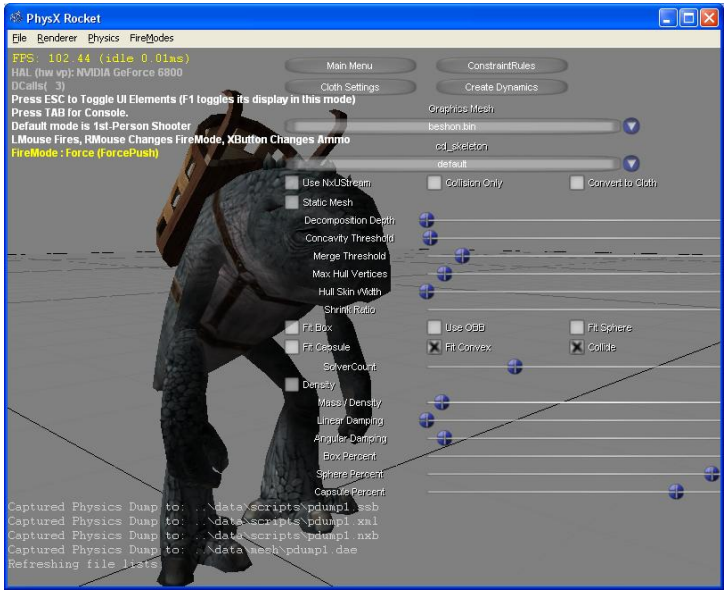
- 3dsMax – Free, Open Source Import/export
- Maya – Free, Open Source Import/export
- XSI – Included, Source included in SDK, import/export
- Blender – OSS DCC tool, Free, Open Source, Import/export
- Houdini – Import only for now
- SketchUp – Free DCC tool, included, export only for now
- …

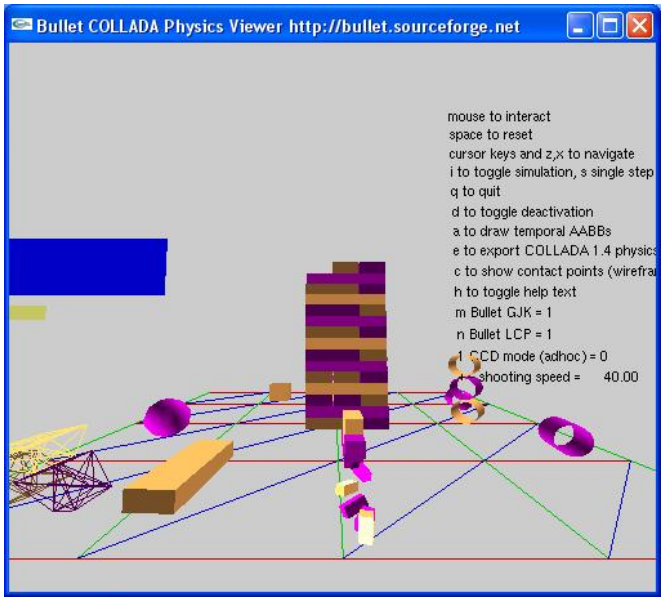
More tools

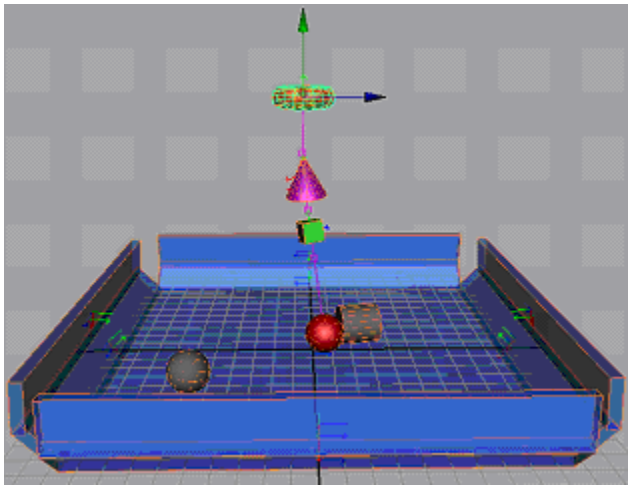
- Feeling Viewer – Free windows binary
- FXComposer 2.0 – To be released (free binary) before end of 2006
- COLLADA-DOM – Reference API, free, open-source
- FCOLLADA – Higher level API, free, open-source
- XSI FTK – API + Viewer, free
- COLLADA-RT – COLLADA run-time/viewer for DOM, free, open-source
- COLLADA-CFX – COLLADA FX for Cg 1.5, free, open-source
- Refinery – COLLADA Content pipeline tool – free, open-source
  - Java GUI, conditioners can be linked in a conditioning pipeline
  - Save and run as batch, or create .exe
  - Triangulation, T-strip, optimization
  - .kmz conditioner
  - Validation and coherency test
- Whoola – Java based viewer, web integration, free, open-source

COLLADA Physics



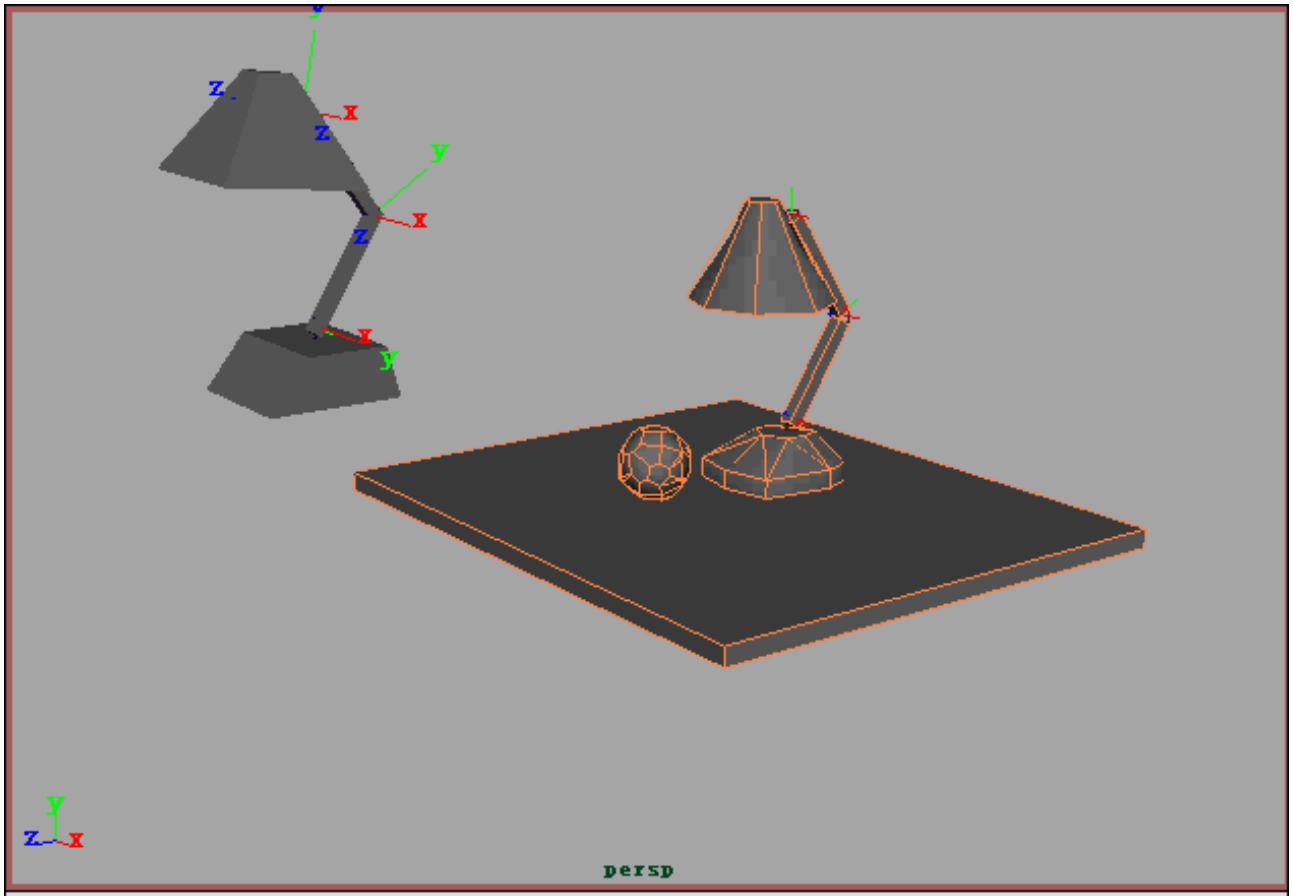






Images courtesy of Feeling Software (www.feelingsoftware.com), Bullet physics library (www.continuousphysics.com/Bullet) and AGEIA (www.ageia.com)

COLLADA Physics and Animation



nima_joint_drive_animation.avi

Content pipeline demo

- (live presentation)

<asset>

Early feedback from the working group study on COLLADA and asset management.

Looking for your feedback – please e-mail us at collada@collada.org

The many definitions of Assets

What is an "Asset" ?

1. Data Storage
2. Version Control
3. Bug tracking
4. Workflow
5. Data Dependency
6. Tool Chain
7. Game assets
8. Asset Composition

Data storage

Data storage, meta-data, and search

- Game developers are notoriously sloppy with their data
  - Nested directory trees
  - Files identified by elaborate naming conventions
- Many data types
  - Images, movies, sound, scripts, shaders, code…

Using a database system seems a better fit than files for assets

Version control

"Asset Management" often is thought of as a synonym of "Version Control"

- Data needs to be backed up
- Data needs to be accessible to everyone
  - Teams need to alter the same data concurrently
  - Opaque data needs locking
- Changes need to be tracked
- Tagging assets with version information
  - Version tags separated from asset is a problem
- Conflicting changes need to be resolved
  - Source code can be merged. Why can't 3D data be merged ?

Bug Tracking

“Version Control” and bug tracking are tightly coupled.

- Who broke the build ?
  - Automatic feedback in auto build systems
- What to do with broken assets ?
  - Using previous version
  - Ignoring broken data
  - Warning to the user

Asset needs bug tracking links

Workflow

assets need to fit within the workflow

- Who has created this asset ?
- Who has modified this asset ?
- Place holder assets
- Is this asset approved ?
  - Access rights management
- What is the copyright ?

Need flexible workflow meta-data

Data Dependencies

Tracking and updating file dependencies

- Assets are hierarchical
  - Geometry + textures + movies + physics + …
  - Libraries
  - Hierarchy is defined per project
- Extracting asset with all dependencies
  - Query, search
  - Create subset of files -> new asset ?

Need to maintain dependency across assets / files

Tool Chain

The build process applies tools on the assets

- Game Data needs processing for run-time consumption
  - Requires Asset Dependency knowledge
- Short cuts are necessary for productivity
  - Feedback loop
  - Fast path implementation issues
- Needs a build system
  - File time stamp method result in major inefficiency
  - Dependency lost in script system
  - Configuration management for multi-platform support

  Build information should be stored with the data

Game assets

Data needs to be tagged and associated into game-specific groups

- Game objects are collections of assets
- A system is needed for grouping this data
  - LODs – Multiple representation
    - Need metadata for run-time selection
  - Base meshes and bounding geometry
  - Visual FX and sound / particle effects
  - Animation sequences

Needs a standard system for tagging and grouping data

Asset Composition

Individual pieces of digital content can be treated as separate "assets"

- A single digital content file may contain many different types of objects
  - Geometry
  - Shaders
  - Animations
- Build tools act on specific asset types
- #1 productivity issue: partial updates
  - Build only what's needed
  - Update instead of export ?
  - Computer Memory limitations

Need to manage asset composition

COLLADA asset

- Definition:
  - An asset in COLLADA is an element with an asset tag
    <element>
    <asset>
- <COLLADA> document must have an <asset> tag
- <asset> are hierarchical and have meta-data such as <unit> <created> <modified> <copyright>
  - A COLLADA document can be composed of several <assets>
- <asset> element contains information about the asset itself (author, date, version, source, unit system…)

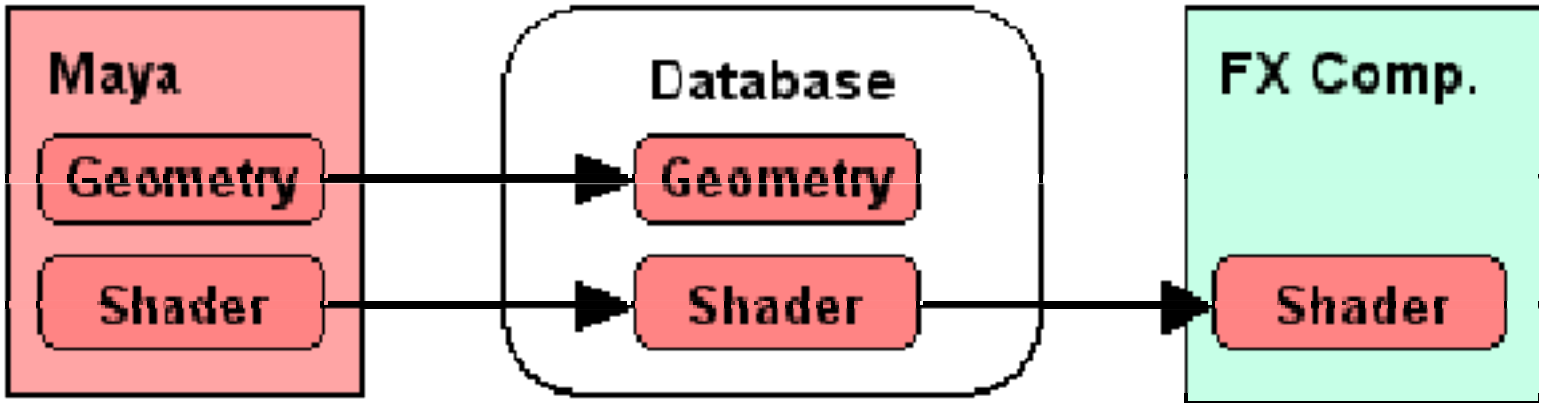
New notion: Asset Ownership

Every asset is owned by a single tool, however a document may contain assets from many owners

- The asset owner holds the original data (example source data in Maya .mb)
- Modification of un-owned assets is forbidden
- Ownership may be transferred between tools
  - Ownership transfer is sometimes necessary
  - May result in data loss or corruption – need user validation
- Exports of un-owned assets will be ignored – need to be exported from source data

Ownership transfer 1/3

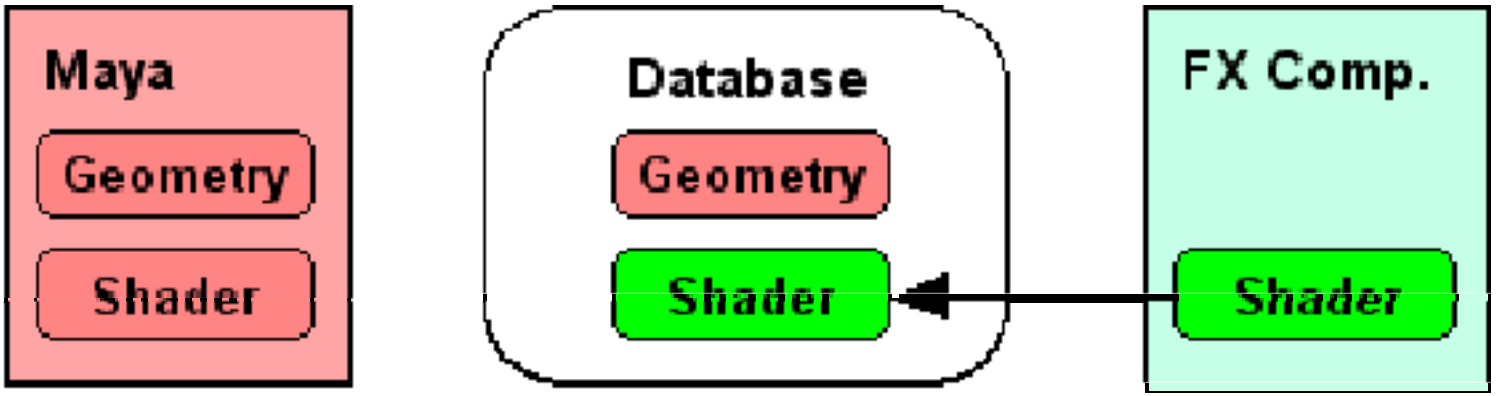
- File created, all assets exported



Each asset type is best edited in a specific tool, ownership can be transferred to that tool

Ownership transfer 2/3

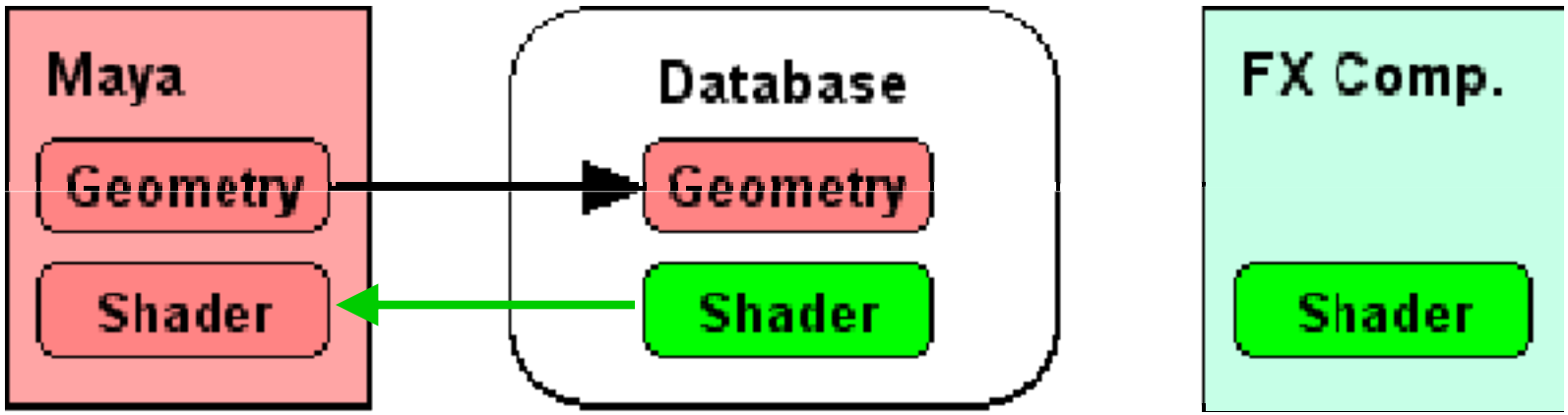
Explicit Asset Ownership Transfer of shader to FXComposer



Explicit asset handoffs are accompanied by warnings. The user must acknowledge that data may be lost in the process

Ownership transfer 3/3

New Maya export, un-owned assets not exported



Once an asset has changed owners, attempted exports from all other tools are ignored.
Maya original version need to be replaced with data from new owner.

Questions ?

www.khronos.org/collada

www.collada.org

collada@collada.org