

# X3D Graphics for Web Authors

## Chapter 1

### Technical Overview

*When we mean to build, we first  
survey the plot, then draw the model.*

William Shakespeare, Henry IV

# Contents

## Chapter Overview

## Technical Introduction

- VRML historical background
- Web3D Consortium
- Browsers and X3D Specifications
- scene graph
- Profiles + components, field and node data types
- XML encoding, ClassicVRML, Compressed binary

## Additional Resources and Chapter Summary

## References and Book testimonials

# Chapter Overview

# Overview: Technical Introduction

This chapter provides a broad overview of how X3D graphics is designed and implemented

- Goal is to provide quick coverage of many features

For newcomers to X3D, a quick read is sufficient

- Getting started building models in Chapters 2 and 3 is more important than understanding every point
- Can review again later to reinforce concepts

Details found in Chapter 1, *X3D for Web Authors*

- This [chapter](#) is available free online

# X3D Technical Introduction

# Historical background: VRML

Virtual Reality Modeling Language (VRML) began in 1994, seeking to create 3D markup for Web

- Numerous candidates considered by an open community of interested practitioners
- SGI's OpenInventor won the initial competition
- VRML 1.0 developed over the next year
- VRML 2.0 restructured some nodes, added features

VRML advanced to International Standard 14772 by ISO in 1997

# Web3D Consortium

Web3D Consortium founded in 1998 to protect, support and advance the VRML specification

- <http://www.web3D.org>

Continued efforts on new technology by multiple working groups led its successor, X3D

- <http://www.web3D.org/x3d>

Non-profit organization of many stakeholders ensures that X3D remains royalty free, relevant

- Partnership of industry, agency, academic and professional members



### Featured Case Study Penn State to visualize cave with VRML/X3D based software

Penn State has developed cave applications, targeting to facilitate the effective use of virtual reality (VR) techniques in design, construction and other disciplines. The aim was to develop immersive environments, where users can study all kinds of designs for different ranges with special visualization techniques. Penn State opted for VRML/X3D- based standard software, BSContact Stereo from Bitmanagement. [More FR](#)

The core technology of the cave applications is the 3D visualization software, BS contact Stereo software, completely meets the requirements of a high quality stereoscopic rendering, which is essential to create a fully immersive environment. BS Contact Stereo is also an interactive, real time and internet ready software, which above all showing the stereo effect without special eyeglasses.

These are visualization systems with two, three or more ceiling high projection walls, which obtain a realistic three-dimensional impression. The stereoscopic effect is generated by two projectors in each wall. Each single wall is synchronized with all others, whereby the position of the watching person is covered by a tracking system, so that the watchers point of view is adapted to his or her respective location and accordingly to his or her moves. By the immersive perception - the stereoscopic 3D view - product can be analyzed virtually, but with visible depth effect.

[Read more](#)



## Latest Web3D News

[Submit News](#)

### Bitmanagement Announces the Release of VRML/X3D Viewer BS Contact 7.1.

Dec 22, 2007 Bitmanagement announces the release of VRML/X3D viewer BS Contact 7.1., and BS Collaborate Multi-User Server.

The BS Collaborate server is based on Web3D's proposal for the Networking component (networkSensor). These nodes allow VRML/X3D scenes to connect to arbitrary servers or directly link between two VRML/X3D players. With these nodes one can manipulate virtual objects collaboratively in real time.

Highlights of this new release:

- \* BS Contact, with it's new BS Collaborate Server, now supports virtual worlds in which multi-user avatar functionality can be utilized.
- \* BS Contact is now "multi-lingual", it supports not only the standard web3D formats X3D and VRML but also COLLADA.
- \* BS Contact is now interfaced to the high-speed Ageia PhysX engine as well as ODE, to aid in the visualisation of realistic simulations.
- \* BS Contact now enables the use of Flash in 3D applications with transparency.
- \* BS Contact with it's new automated installation mechanism is noticeably easier to load and start.

Many of these new features open the door for more applications and allows users to capitalize on the current trend towards interactive virtual worlds. You can download a test version at: <http://www.bitmanagement.com/download/playerdownload.en.html>

<http://www.bitmanagement.com/documents/Pressemappe/BS Contact 7.1 E.pdf>

<http://www.bitmanagement.com>

Category: Applications | [Permalink](#)

### SIGGRAPH 2007 - Web3D's "Tech Talk" Podcast Now Available

Dec 19, 2007 SIGGRAPH 2007 - Web3D's "Tech Talk" Podcast

## Member Login

  
 Username:  
  
 Password:  


[Not yet a member?](#)

## Recent Blog Posts

Virtual Worlds Today: A really distortion field in the making?

We would like to thank all visitors to the Web3D booth at Siggraph!

Is Siggraph now Hollywood's version of E3?

Web3D Executive Director's Summer and Siggraph 2007 Update

Some tips for getting around on Web3D.org and getting your questions on X3D answered

[View all blog entries](#)





### Featured Case Study

#### ALIVE, University of Southern Queensland - The Phoenix Challenge

ALIVE created "The Phoenix Challenge" using Flux Studio, Maya and Rawkee and implemented Ajax3D to keep track of the player's score and to interact with their database of objects. The player's objective is to make their way around the campus picking up objects which affect their STRENGTH, SMARTS and STRESS. There is a time limit to complete each level. X3D provided the ability to create a browser based game which students can find by following a URL and after installing the Flux Player, do not require other software on their system and can view other X3D scenes the ALIVE team create.

The game can be played by visiting <http://www.alivex3d.org/challenge>

Viewers can access this password protected game with the username: guest and the password: guest

[Read more](#)



## Latest Web3D News

[Submit News](#)

### Fraunhofer IGD releases Beta4 of the InstantReality System

Jan 21, 2008 Included in this release are the following new and/or improved features along with minor and major bugfixes.

- One of the first (alpha/ beta) implementations of the X3D Med/ VolumeRendering Component specification.
- The GPU based ray-caster allows to visualize volume-data very efficiently.
- Faster and more robust automatic optimization of static subtrees.
- Improved support for StaticGroup and Inlines of static data
- Improved cluster synchronization mechanisms. It is now much easier to setup a stereo wall or even a CAVE
- Improved cluster performance, especially if local and remote windows are used.
- Multi-touch extensions for all PointingSensor-nodes
- Geometry Shader extensions for the shader node-sets. Only available on graphics wahardre with Shader Model 4.0 (e.g. NVidia 8x).

The new labs section of the web-page <http://www.instantreality.org/labs> provides documentation, tutorials and neat tools to convert classic/xml data.



### Member Login

Username:

Password:

Not yet a member?

### Recent Blog Posts

Virtual Worlds Today: A really distortion field in the making?

We would like to thank all visitors to the Web3D booth at Siggraph!

Is Siggraph now Hollywood's version of E3?

Web3D Executive Director's Summer and Siggraph 2007 Update

Some tips for getting around on Web3D.org and getting your questions on X3D answered

[View all blog entries](#)

Become an X3D blog author!

# X3D browsers

X3D browsers parse (read) X3D scene models and render (draw) them

- Also provide simulation capabilities for animation and user interaction

Often implemented as plugins to web browsers:

- Internet Explorer <http://www.microsoft.com>
- Mozilla Firefox <http://www.mozilla.com>
- Opera <http://www.opera.com>

Can also operate as a standalone application

- Xj3D <http://www.xj3d.org>

## Example software architecture for X3D browser

3D graphics algorithms and implementations are intensely technical and performance-sensitive

X3D browsers are thus allowed to implement in any manner which they choose

- As long as the author's X3D scene works properly

This is a healthy split of responsibilities

- Each gets to excel at what they are good at

Commonalities and shared lessons learned continue to build up nicely

- Next diagram shows example architecture

X3D scenes,  
X3D streams

Event passing with external  
HTML Web pages or applications

## X3D Browser

### Parsers

X3D XML  
encoding

Classic VRML  
encoding

Binary  
encoding

### Scene Access Interface (SAI)

Application programmer interfaces

### New node and prototype construction

X3D  
nodes, node types

Prototype and  
External Prototype

Scene graph manager

### Scripting engines

EcmaScript  
Java  
others

Scene Graph Renderable Nodes

Event Graph Animation Nodes

# X3D Specifications

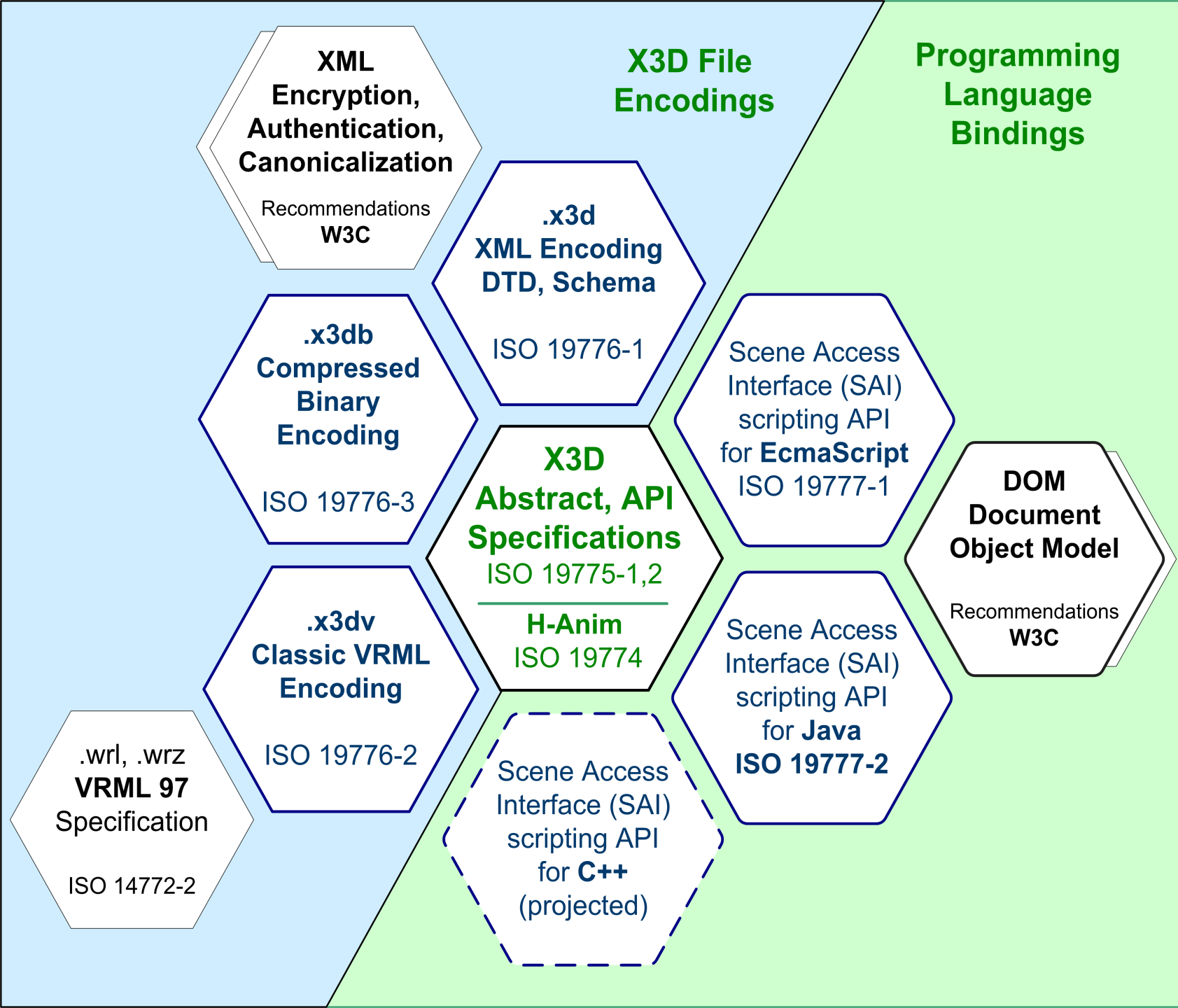
X3D graphics is defined by a set of specifications

These “specs” are developed by working-group volunteers as part of the Web3D Consortium

- Nonprofit organization with business, nonprofit, academic and professional members
- <http://www.web3D.org>
- Efforts include editing, implementing and evaluating

Specification results reviewed and approved by International Organization of Standards (ISO)

- <http://www.iso.ch>



# Reading the X3D specification

The X3D Specification is highly detailed, primarily written for 3D graphics experts.

Requirements must be described as strictly and precisely as possible so that X3D browsers can be implemented consistently. This precision means that X3D content is more likely to render and animate correctly.

Nevertheless the X3D specification is a great learning resource for additional graphics details. It is also the authoritative reference for questions.

# Specification availability

The X3D specifications are online at

- <http://www.web3d.org/x3d/specifications>
- also embedded in the X3D-Edit help system

The X3D specifications are published by the Web3D Consortium and International Organization of Standards (ISO)

- Web3D versions are published in HTML for free
- ISO publishes .pdf versions and requires purchase

Feedback on X3D specifications is always welcome

- [http://www.web3d.org/x3d/specifications/spec\\_feedback](http://www.web3d.org/x3d/specifications/spec_feedback)



# Scene graph concepts

Scene graphs are a model-centric approach to 3D that hierarchically defines geometry shape, appearance, position and orientation, etc. etc.

- Directed acyclic graph (DAG), meaning a tree with a root node and no loops
- Declarative listing of parameters of interest
- Similar to defining a Computer Aided Design (CAD) model

Unlike most imperative programming approaches

- draw this triangle, that triangle, recompute, etc.

# Behaviors

*Behavior* defined as changing the value of some field contained by some node in scene graph

Animation nodes, user interaction nodes and network updates can produce updated values

ROUTE statements connect output of one node as an input to field in another node

*Event* defined as the time-stamped value passed by a ROUTE, from one field to another

Thus the values held by nodes in scene graph can change as time advances

# Scene graph rendering

The browser traverses the scene graph, updating any values within nodes and building an image

- New image then replaces previous screen image, process known as *double buffering*
- Rapid repetitions are very important
- Frame rate faster than 7-10 Hz (cycles per second) provides appearance of smooth motion

*Rendering* defined as this drawing process

*Off-line rendering* is performing such operations to image or movie files, rather than display

# X3D file structure

X3D scene files have a common file structure

- File header (XML, ClassicVRML, Compressed Binary)
- X3D header statement
- Profile statement
- Component statements (optional)
- Meta statements (optional)
- X3D root node
- X3D scene graph child nodes

# Need for subdivisions and subsets

3D graphics is a big and complicated subject

- Beginning authors just want simple scenes
- Experienced authors want to use everything

Similar needs for browser software builders

- Small rapid download for simple web graphics
- Full-capability software for every possible technique

Challenge: how to consistently support both?

- Object-oriented decomposition for consistency
- Key design criteria for bottom-up X3D extensibility

# Profiles and components

Profiles are predefined collections of components

- Can augmented each by adding other components

Components are predefined collections of nodes

- Further defined by *level* of complexity
- Components match chapters in X3D specification

Authors define the expected complexity of scene by defining profile level in the X3D header

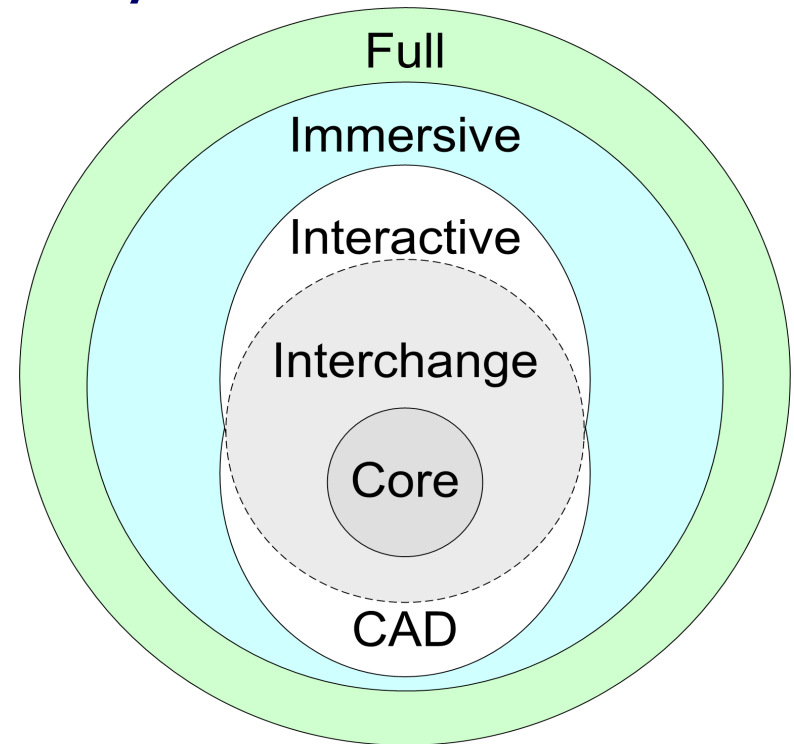
- Can also add optional components, if desired
- This tells the X3D browser what level of support is needed for run-time operation

# Profiles cover common use cases

Profiles are a collection of components matching common levels of complexity

Profiles are X3D subsets

- Collection of X3D nodes for author's palette
- *Interchange* suitable for simple geometry conversion
- *Interactive* adds simple user interactivity (clicking etc.)
- Immersive matches VRML97
- Full profile includes all nodes



# meta statements

meta statements provide information about the X3D scene

- Document metadata, not scene metadata

Information provided as name-value pairs

- Example:

```
<meta name='created' value='1 January 2008' />
```

This approach is thus very general

- Wide variety of metadata can be represented
- Matches same approach used by HTML for regular hypertext web pages



# profile, component and meta statements, XML (.x3d) encoding syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.2.dtd">
<X3D version="3.2" profile="Immersive" xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance"
      xsd:noNamespaceSchemaLocation="http://www.web3d.org/specifications/x3d-3.2.xsd">
  <head>
    <component name='DIS' level='1'/>
    <component name='Geospatial' level='1'/>
    <component name='H-Anim' level='1'/>
    <component name='NURBS' level='4'/>
    <meta name='title' content='HeaderProfileComponentMetaExample.x3d'/>
  </head>
  <Scene>
    <!--Scene graph nodes are added here-->
  </Scene>
</X3D>
```

# Profile, component and meta statements, ClassicVRML (.x3dv) encoding syntax

#X3D V3.2 utf8

**PROFILE** Immersive

# No HEAD statement is provided in ClassicVRML Encoding

**COMPONENT** DIS:1

**COMPONENT** Geospatial:1

**COMPONENT** H-Anim:1

**COMPONENT** NURBS:4

**META** "filename" "HeaderProfileComponentMetaExample.x3d"

# Scene graph nodes are added here

# newScene.x3d metadata prompts

<meta content='\*enter FileNameWithNoAbbreviations.x3d here\*' name='title'/>

<meta content='\*enter description here, short-sentence summaries preferred\*' name='description'/>

<meta content='\*enter name of original author here\*' name='creator'/>

<meta content='\*if manually translating VRML-to-X3D, enter name of person translating here\*' name='translator'/>

<meta content='\*enter date of initial version here\*' name='created'/>

<meta content='\*enter date of translation here\*' name='translated'/>

<meta content='\*enter date of latest revision here\*' name='modified'/>

<meta content='\*enter version here, if any\*' name='version'/>

<meta content='\*enter reference citation or relative/online url here\*' name='reference'/>

<meta content='\*enter additional url/bibliographic reference information here\*' name='reference'/>

<meta content='\*enter reference resource here if required to support function, delivery, or coherence of content\*' name='requires'/>

<meta content='\*enter copyright information here\* Example: Copyright (c) Web3D Consortium Inc. 2008' name='rights'/>

<meta content='\*enter drawing filename/url here\*' name='drawing'/>

<meta content='\*enter image filename/url here\*' name='image'/>

<meta content='\*enter movie filename/url here\*' name='MovingImage'/>

<meta content='\*enter photo filename/url here\*' name='photo'/>

<meta content='\*enter subject keywords here\*' name='subject'/>

<meta content='\*enter permission statements or url here\*' name='accessRights'/>

<meta content='\*insert any known warnings, bugs or errors here\*' name='warning'/>

<meta content='\*enter online Uniform Resource Identifier (URI) or Uniform Resource Locator (URL) address for this file here\*' name='identifier'/>

<meta content='X3D-Edit, <https://savage.nps.edu/X3D-Edit>' name='generator'/>

<meta content='../..//license.html' name='license'/>

# Field data types

X3D is a strongly typed language

- Each field in each node (i.e. each XML attribute) has a strictly defined data type
- Data types for boolean, integer, floating point

Types are either single or multiple-value

- Example: SFFloat, SFVec2f, SFVec3f, SFOrientation

Also have arrays for all types

SF = Single Field, MF = Multiple Field (array)

**Failure to match data types correctly is an error!**

- During schema validation, loading or at run time

# Field data types

Field-type names	Description	Example values
SFBool	Single-field boolean value	true or false (X3D syntax), TRUE or FALSE (ClassicVRML syntax)
MFBool	Multiple-field boolean array	true false false true (X3D syntax), [ TRUE FALSE FALSE TRUE ] (ClassicVRML syntax)
SFColor	Single-field color value, red-green-blue	0 0.5 1.0
MFColor	Multiple-field color array, red-green-blue	1 0 0, 0 1 0, 0 0 1
SFColorRGBA	Single-field color value, red-green-blue alpha (opacity)	0 0.5 1.0 0.75
MFColorRGBA	Multiple-field color array, red-green- blue alpha (opacity)	1 0 0 0.25, 0 1 0 0.5, 0 0 1 0.75 (red green blue, varying opacity)
SFInt32	Single-field 32-bit integer value	0
MFInt32	Multiple-field 32-bit integer array	1 2 3 4 5
SFFloat	Single-field single-precision floating- point value	1.0
MFFloat	Multiple-field single-precision floating- point array	-1 2.0 3.14159

# Field data types

Field-type names	Description	Example values
SFDouble	Single-field double-precision floating-point value	2.7128
MFDouble	Multiple-field double-precision array	−1 2.0 3.14159
SFImage	Single-field image value	Contains special pixel-encoding values, see Chapter 5 for details
MFImage	Multiple-field image value	Contains special pixel-encoding values, see Chapter 5 for details
SFNode	Single-field node	<Shape/> or Shape {space}
MFNode	Multiple-field node array of peers	<Shape/><Group/><Transform/>
SFRotation	Single-field rotation value using 3-tuple axis, radian angle form	0 1 0 1.57
MFRotation	Multiple-field rotation array	0 1 0 0, 0 1 0 1.57, 0 1 0 3.14
SFString	Single-field string value	"Hello world!"
MFString	Multiple-field string array	"EXAMINE" "FLY" "WALK" "ANY"
SFTime	Single-field time value	0
MFTime	Multiple-field time array	−1 0 1 567890

# Field data types

Field-type names	Description	Example values
SFVec2f/SFVec2d	Single-field 2-float/2-double vector value	0 1.5
MFVec2f/MFVec2d	Multiple-field 2-float/2-double vector array	1 0, 2 2, 3 4, 5 5
SFVec3f/SFVec3d	Single-field vector value of 3-float/ 3-double values	0 1.5 2
MFVec3f/MFVec3d	Multiple-field vector array of 3-float/ 3-double values	10 20 30, 4.4 –5.5 6.6

## ClassicVRML syntax notes

- TRUE and FALSE (rather than XML true and false)
- MF multiple-field array values are surrounded by square brackets, e.g. [ 10 20 30, 4.4 –5.5 6.6 ]
- No special XML escape characters such as **&amp;**;

# accessType: input, output, initialize

accessType determines if field is data sender, receiver, or holder

- inputOnly: can only receive events
- outputOnly: can only send events
- initializeOnly: cannot send or receive
- inputOutput: can send, receive and be initialized

**Failure to match accessType correctly is an error!**

- Detected during authoring-tool checks, or run time



# accessType naming conventions

The accessType names were changed when VRML97 was upgraded to X3D

- Functionality remains essentially unchanged

X3D specification entries for each node use yet another shorthand, as shown here

VRML97 Name	X3D Name	X3D Specification abbreviation
eventIn	inputOnly	[in]
eventOut	outputOnly	[out]
field	initializeOnly	[ ]
exposedField	inputOutput	[in,out]
VRML, Virtual reality modeling language; X3D, Extensible 3D.		

# Abstract node types

## X3D nodes also have strong typing

- Provides consistent field interfaces for similar nodes
- Object-oriented improvement over VRML97, which had several internal inconsistencies
- Better language design

## Benefits include

- Allowed child-node content is consistent
- Simple-type field values have identical defaults
- Application programming interfaces more consistent
- Definitions are easier to remember and apply

# XML file encoding

The Extensible Markup Language (XML) is a plain-text format used by many Web languages

- Including Hypertext Markup Language (HTML)

XML is used to define other data-oriented languages

- Thus XML is not a language by itself, rather it is a language about languages, a *metalanguage*

XML has many benefits and is well-suited for X3D

# XML in 10 Points

<http://www.w3.org/XML/1999/XML-in-10-points>

XML is for structuring data

XML looks a bit like HTML

XML is text, but isn't meant to be read

XML is verbose by design

XML is a family of technologies

XML is new but not that new

XML leads HTML to XHTML

XML is modular

XML is basis for RDF and the Semantic Web

XML is license-free,  
platform-independent and  
well-supported

*XML in 10 Points* is a key reference for understanding the common underlying design principles underlying the great diversity of XML.

Only 4 pages long – essential reading.

# XML and X3D correspondence

Opening element  
Singleton element, `attribute="value"`  
Opening element  
Singleton element, `attribute='value'`  
Closing element  
Closing element

```
<Shape>  
  <Sphere radius="10.0" solid="true"/>  
  <Appearance>  
    <ImageTexture url='earth-topo.png'/>  
  </Appearance>  
</Shape>
```

Elements correspond to X3D nodes

Attributes correspond to X3D simple-type fields

Parent-child relationships define containerField

Validatable XML using X3D DTD, schema

# XML validation

XML validation applies XML rules to an XML document to confirm whether it is correct

- *Well formed XML*: legal header, matching open/close tags, proper attribute-value pairs, etc.
- *DTD (DOCTYPE) validation*: adds checks on legal element and attribute names, proper parent-child relationships, simple checks on attribute values
- *XML Schema validation*: also includes stricter datatype-aware checks on attribute values

XML validation finds problems before end users

- reducing garbage-in garbage-out (GIGO)

# ClassicVRML file encoding

The ClassicVRML file syntax is a direct, backwards-compatible extension of VRML97

- VRML version 2.0 became X3D version 3.0, 3.1 etc.
- No changes in syntax rules
- Some additional new nodes and slight naming differences to match specification improvements
- VRML97 content still works and is easily supported

XML, ClassicVRML and Compressed Binary encodings are functionally equivalent

- Governed by same X3D abstract specification

# Compressed binary encoding

Two types of compression for .x3db encoding

- XML-centric ISO Fast Infoset
- Geometry-centric for coplanar polygons, quantization of points, colors & normals, etc.

Java3D algorithms are default for geometry compression

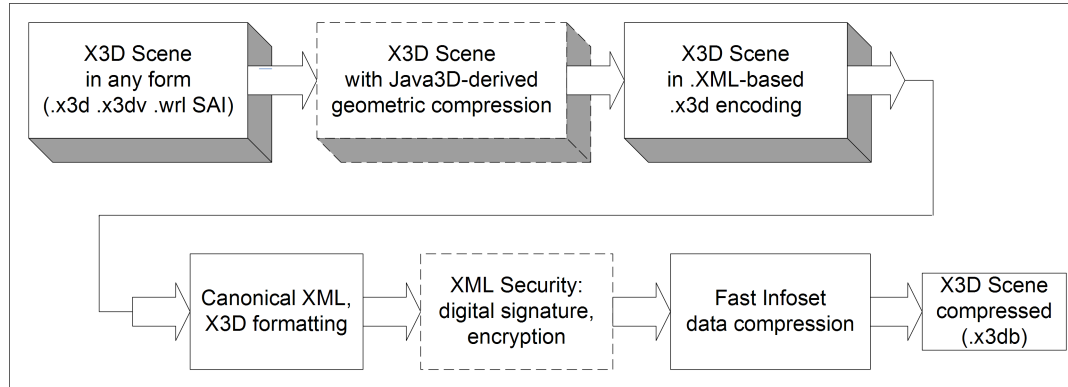
- Royalty free for use with X3D
- Other uses – please contact Sun Microsystems

Alternate geometry compression is allowed

Sample implementation: Xj3D



# X3D compressed binary algorithm and XML Security



X3D compressed binary uses Canonical X3D form

- Strict formatting rules so that files with identical format can be shown to match

Canonical form enables use of XML Security

- XML Encryption
- XML Digital Signature (for author authentication)

# Additional Resources



Open Standards for Real-Time 3D Communication

[New Web3D Visitor?](#) | [Join the Consortium](#) | [Contact Us](#) | [Calendar](#) | [Site Map](#)

Google Custom Search

[Home](#) [About Web3D](#) [X3D Development](#) [News & Events](#) [Community](#) [Membership](#)

[Home](#) » [X3D Development](#) » [Public Mailing Lists](#)

## Public Mailing Lists

Consortium [Members](#) have access to all internal X3D Technical and Working Group mailing lists including X3D, UI, Med, CAD, Conformance, DIS-XML, and Shaders.

Non-members are invited to use the [X3D user-to-user message boards](#) and [public emails lists](#) listed below

### X3D-Public Email list

The X3D mailing list is the place to to have focused discussions and feedback on X3D technical and development issues. It is appropriate for content creators, end-users and coders . This list is the primary vehicle non-members to communicate with and receive announcements from the Web3D Consortium board and X3D Specification team.

**Online Sign Up** - To subscribe or unsubscribe to the X3D-Public email list use the [online interface](#).

**Manual Sign Up** - To manually subscribe to the list, send an email to [majordomo@web3d.org](mailto:majordomo@web3d.org). In the body of your message include the line *subscribe x3d-public*. You will receive back a confirming email when your subscription has been approved. To unsubscribe send an email to [majordomo@web3d.org](mailto:majordomo@web3d.org). In the body of your message include the line *unsubscribe x3d-public*.

**X3D-Public List Archives** - Browse through the [archives](#) of past X3D-Public list messages.

### WWW-VRML Public Email list

The VRML mailing list is the place to discuss VRML97 (the predecessor to X3D) issues, philosophies and application. The comments on this list represent the views and knowledge of the comment authors only. They do not necessarily represent official views from the Consortium.

**Online Sign Up** - To subscribe or unsubscribe to the WWW-VRML email list use the [online interface](#).

**Manual Sign Up** - To manually subscribe to the list, send an email to [majordomo@web3d.org](mailto:majordomo@web3d.org). In the body of your message include the line *subscribe www-vrml*. You will receive back a confirming email when your subscription has been approved. To unsubscribe send an email to [majordomo@web3d.org](mailto:majordomo@web3d.org). In the body of your message include the line *unsubscribe www-vrml*.

**WWW-VRML List Archives** - Browse through the [archives](#) of past WWW-VRML list messages.

### Web3D-Announce Marketing and News Newsletter

[Sign up](#) for the newsletter.

#### Member Login

Username:  
  
Password:

[Not yet a member?](#)

#### Community

- [X3D Case Studies](#)
  - [Submit your Case Study](#)
- [X3D Community Blog](#)
  - [Archives](#)
  - [Add your Voice](#)
  - [RSS feed](#)
- [Sample X3D Models](#)
  - [Submit your X3D Models](#)
- [X3D Wiki](#)
- [Developer Message Boards](#)
- [Public Email Lists](#)
- [Community Newsletters](#)

# Annual Conferences

## SIGGRAPH

- <http://www.siggraph.org>

## Web3D Symposium

- <http://www.web3d.org/conferences/web3d2008>
- Web3D 2007 Tech Talk podcast

## Eurographics

- <http://www.eg.org>

# Web3D liaison organizations

## World Wide Web Consortium (W3C)

- Leading the Web to its Full Potential
- <http://www.w3.org>

## Open Geospatial Consortium

- Leading the development of standards for geospatial and location-based services.
- <http://www.opengeospatial.org>

# Chapter Summary

# Chapter Summary

This technical overview chapter is a mile wide and a few meters deep. Key points include

- VRML historical background
- Web3D Consortium
- Browsers, X3D Specifications, scene graph
- Profiles + components, field and node data types
- XML encoding, ClassicVRML, Compressed binary

New students of X3D can refer to details later.

Get working on examples in the next chapters!

# References



# References 1

*X3D: Extensible 3D Graphics for Web Authors*  
by Don Brutzman and Leonard Daly, Morgan  
Kaufmann Publishers, April 2007, 468 pages.

- Chapter 3, Grouping Nodes
- <http://x3dGraphics.com>
- <http://x3dgraphics.com/examples/X3dForWebAuthors>

## X3D Examples Help

- <http://www.web3d.org/x3d/content/examples/help.html>

# References 2

## X3D Scene Authoring Hints

- <http://x3dgraphics.com/examples/X3dSceneAuthoringHints.html>

## X3D Graphics Specification

- <http://www.web3d.org/x3d/specifications>
- Also available as help pages within X3D-Edit

# References 3

*Computer Graphics, Principles and Practice*, by James D. Foley, Andries van Dam, Stephen K. Feiner and John F. Hughes, Addison-Wesley, 2nd Edition, 1997.

Bert Bos et al., "XML in 10 Points," World Wide Web Consortium (W3C), created 1999, updated 2003. Available at <http://www.w3.org/XML/1999/XML-in-10-points>

# Book testimonials 1

There will be no problem understanding these concise, clear, comprehensible background concepts for readers new to Extensible 3D (X3D). There are many notes and examples that compare X3D to Virtual Reality Modeling Language (VRML) features. Don Brutzman and Leonard Daly clearly and thoroughly illustrate each logical concept and feature of X3D with diagrams, tables, code snippets, screenshots of 3D objects/environments, and example scenes, while making use of the very latest specifications and implementations. Their approach contributes greatly to an easy and in-depth understanding of the X3D language. This book is the ultimate introductory guide to X3D!

—Dr. Vladimir Geroimenko, University of Plymouth,  
School of Computing Communications and Electronics, Plymouth, UK

# Book testimonials 2

This book is required reading for anybody interested in Web3D. The authors are well known and respected in the X3D community as pioneers. Their writing style is concise and engaging, set at an appropriate level to encourage understanding, and uses the concepts being introduced. Their “Hints and warnings” sections provide added value above what is available from X3D specification documents. Hard to achieve in a reference manual!

—Professor Nigel W. John, School of Computer Science,  
University of Wales, Bangor; Chair of Web3D 2005 Symposium

# Book testimonials 3

How many times have we heard “The ISO specification is hard to read, do you have something more approachable?” This book is the answer. It provides a detailed explanation of each node in the Immersive profile and gives many reusable examples. After reading this book you’ll be well prepared to develop your own X3D content.

—Alan Hudson, President Web3D Consortium, Yumetech Inc.

This is a much-needed book about the X3D standard and X3D content development. The book follows the structure of the X3D standard specifications which helps readers understand and apply the X3D standard. It can also be used as a reference material in virtual reality and graphics-related courses.

—Professor Denis Gracanin, Virginia Polytechnic Institute & State University, Chair Web3D 2006 Symposium

# Contact

**Don Brutzman**

*[brutzman@nps.edu](mailto:brutzman@nps.edu)*

*<http://web.nps.navy.mil/~brutzman>*

Code USW/Br, Naval Postgraduate School  
Monterey California 93943-5000 USA

1.831.656.2149 voice

1.831.656.7599 fax

# Open-source license

Copyright (c) 1995-2008 held by the author(s). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the Naval Postgraduate School (NPS) Modeling Virtual Environments and Simulation (MOVES) Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



# X3D Graphics for Web Authors

## Chapter 1

### Technical Overview

*When we mean to build, we first  
survey the plot, then draw the model.*

William Shakespeare, Henry IV



# Contents

## Chapter Overview

## Technical Introduction

- VRML historical background
- Web3D Consortium
- Browsers and X3D Specifications
- scene graph
- Profiles + components, field and node data types
- XML encoding, ClassicVRML, Compressed binary

## Additional Resources and Chapter Summary

## References and Book testimonials



# Chapter Overview



# Overview: Technical Introduction

This chapter provides a broad overview of how X3D graphics is designed and implemented

- Goal is to provide quick coverage of many features

For newcomers to X3D, a quick read is sufficient

- Getting started building models in Chapters 2 and 3 is more important than understanding every point
- Can review again later to reinforce concepts

Details found in Chapter 1, *X3D for Web Authors*

- This [chapter](#) is available free online



[http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter01-TechnicalOverview/Chapter01-Technical\\_Overview.pdf](http://x3dgraphics.com/examples/X3dForWebAuthors/Chapter01-TechnicalOverview/Chapter01-Technical_Overview.pdf)

[back to Table of Contents](#)

# X3D Technical Introduction



## Historical background: VRML

Virtual Reality Modeling Language (VRML) began in 1994, seeking to create 3D markup for Web

- Numerous candidates considered by an open community of interested practitioners
- SGI's OpenInventor won the initial competition
- VRML 1.0 developed over the next year
- VRML 2.0 restructured some nodes, added features

VRML advanced to International Standard 14772 by ISO in 1997



Lots more can be said here. Indeed numerous books have been written about VRML.

# Web3D Consortium

Web3D Consortium founded in 1998 to protect, support and advance the VRML specification

- <http://www.web3D.org>

Continued efforts on new technology by multiple working groups led its successor, X3D

- <http://www.web3D.org/x3d>

Non-profit organization of many stakeholders ensures that X3D remains royalty free, relevant

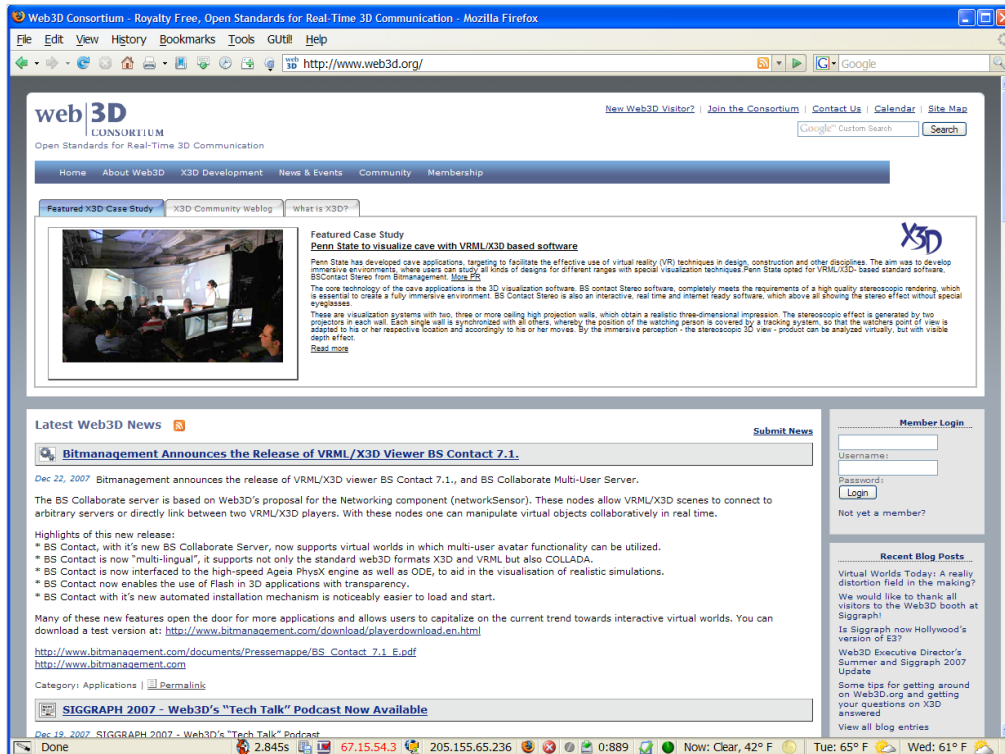
- Partnership of industry, agency, academic and professional members



Perhaps the key test of 'openness' for any self-proclaimed 'open' organization: exactly who is allowed to join? Many industry associations only allow preselected (usually paying) companies to participate.

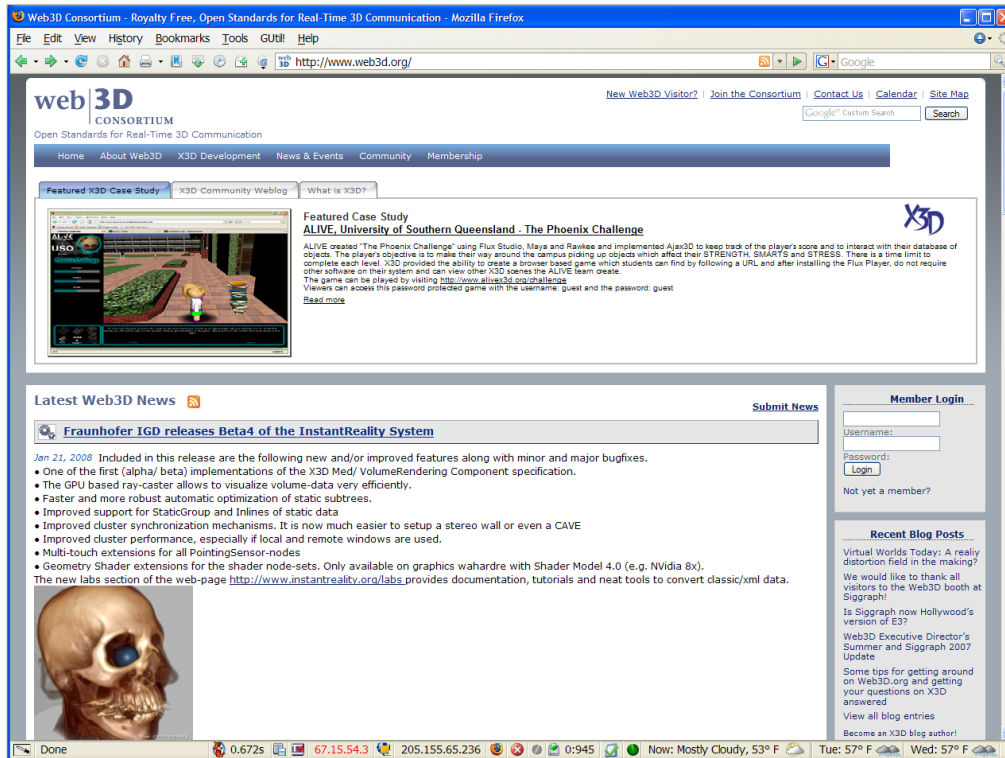
The Web3D Consortium includes industry, government-agency, college/university and individual professional memberships. This makes it one of the most open organizations around.

Further information on membership and joining available online at <http://www.web3d.org/membership>



<http://www.web3d.org> December 2007





<http://www.web3d.org> January 2008

# X3D browsers

X3D browsers parse (read) X3D scene models and render (draw) them

- Also provide simulation capabilities for animation and user interaction

Often implemented as plugins to web browsers:

- Internet Explorer <http://www.microsoft.com>
- Mozilla Firefox <http://www.mozilla.com>
- Opera <http://www.opera.com>

Can also operate as a standalone application

- Xj3D <http://www.xj3d.org>



## Example software architecture for X3D browser

3D graphics algorithms and implementations are intensely technical and performance-sensitive  
X3D browsers are thus allowed to implement in any manner which they choose

- As long as the author's X3D scene works properly

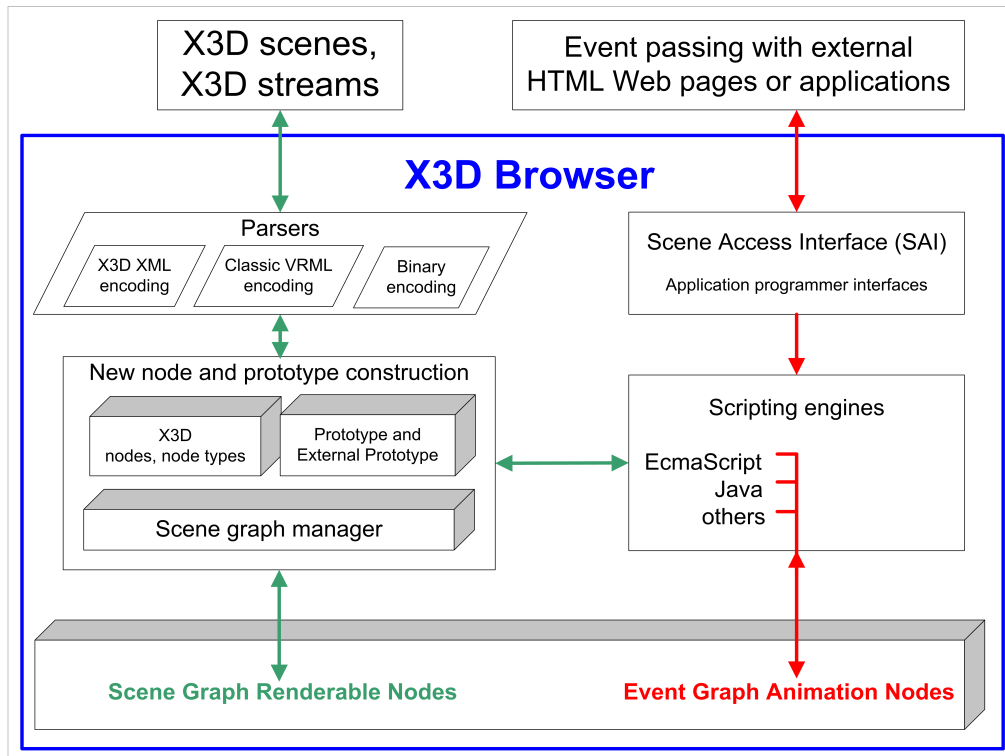
This is a healthy split of responsibilities

- Each gets to excel at what they are good at

Commonalities and shared lessons learned continue to build up nicely

- Next diagram shows example architecture





X3D browser implementers can use any approach they choose. This architecture diagram is generic to illustrate common approaches.

Part of the magic for X3D scene authors is that they don't have to care about underlying hard-core technical details “under the hood” of each browser. Rather, scenes are designed to capture shapes, appearance and behaviors from a content-authoring perspective that emphasizes modeling results.

# X3D Specifications

X3D graphics is defined by a set of specifications  
These “specs” are developed by working-group  
volunteers as part of the Web3D Consortium

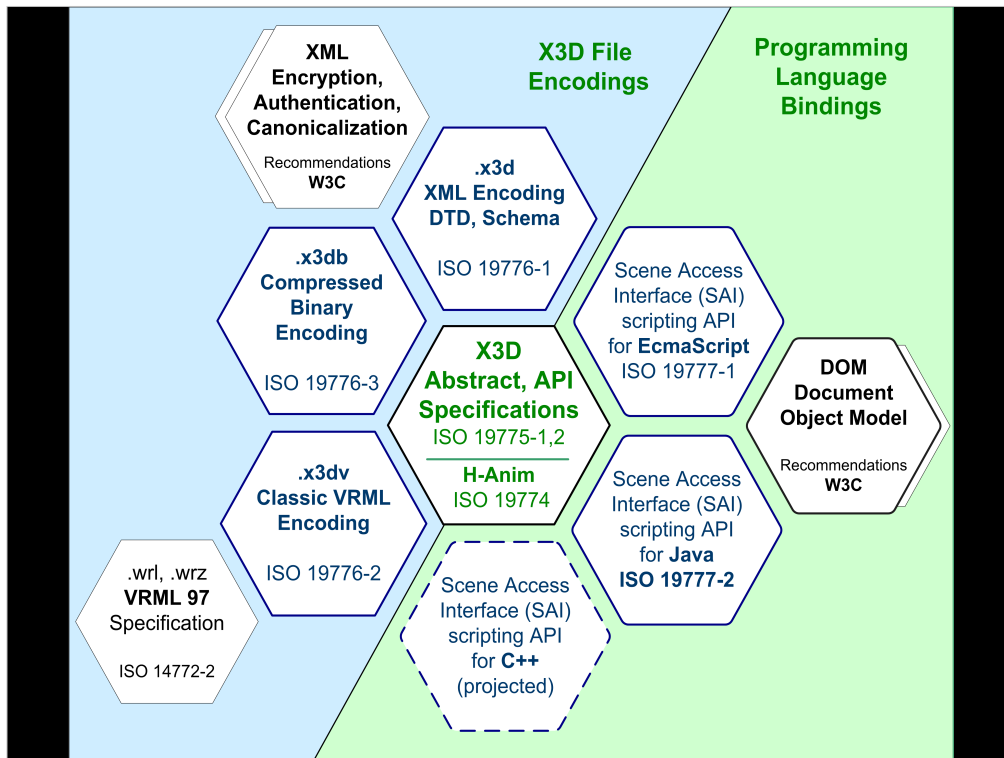
- Nonprofit organization with business, nonprofit, academic and professional members
- <http://www.web3D.org>
- Efforts include editing, implementing and evaluating

Specification results reviewed and approved by  
International Organization of Standards (ISO)

- <http://www.iso.ch>



Typically 10-15 member nations review and vote on the X3D Specification



Encodings define file formats.

Each Scene Access Interface (SAI) binding is a specific Application Programming Interface (API).

ECMAScript is the formal-specification name for JavaScript.

ECMA was originally named the European Computer Manufacturers Association and is now ECMA International - European association for standardizing information and communication systems. <http://www.ecma-international.org>

# Reading the X3D specification

The X3D Specification is highly detailed, primarily written for 3D graphics experts.

Requirements must be described as strictly and precisely as possible so that X3D browsers can be implemented consistently. This precision means that X3D content is more likely to render and animate correctly.

Nevertheless the X3D specification is a great learning resource for additional graphics details. It is also the authoritative reference for questions.



## Specification availability

The X3D specifications are online at

- <http://www.web3d.org/x3d/specifications>
- also embedded in the X3D-Edit help system

The X3D specifications are published by the Web3D Consortium and International Organization of Standards (ISO)

- Web3D versions are published in HTML for free
- ISO publishes .pdf versions and requires purchase

Feedback on X3D specifications is always welcome

- [http://www.web3d.org/x3d/specifications/spec\\_feedback](http://www.web3d.org/x3d/specifications/spec_feedback)



The Web3D Consortium was the first organization to request (and receive) permission to place final versions of approved ISO specifications online for free retrieval using HTML. Purchase of hard-copy bound and electronic versions from ISO remains available.



# Scene graph concepts

Scene graphs are a model-centric approach to 3D that hierarchically defines geometry shape, appearance, position and orientation, etc. etc.

- Directed acyclic graph (DAG), meaning a tree with a root node and no loops
- Declarative listing of parameters of interest
- Similar to defining a Computer Aided Design (CAD) model

Unlike most imperative programming approaches

- draw this triangle, that triangle, recompute, etc.

# Behaviors

*Behavior* defined as changing the value of some field contained by some node in scene graph

Animation nodes, user interaction nodes and network updates can produce updated values

ROUTE statements connect output of one node as an input to field in another node

*Event* defined as the time-stamped value passed by a ROUTE, from one field to another

Thus the values held by nodes in scene graph can change as time advances



# Scene graph rendering

The browser traverses the scene graph, updating any values within nodes and building an image

- New image then replaces previous screen image, process known as *double buffering*
- Rapid repetitions are very important
- Frame rate faster than 7-10 Hz (cycles per second) provides appearance of smooth motion

*Rendering* defined as this drawing process

*Off-line rendering* is performing such operations to image or movie files, rather than display



# X3D file structure

X3D scene files have a common file structure

- File header (XML, ClassicVRML, Compressed Binary)
- X3D header statement
- Profile statement
- Component statements (optional)
- Meta statements (optional)
- X3D root node
- X3D scene graph child nodes



The X3D scene root node is implicit in ClassicVRML encoding and not listed per se.

# Need for subdivisions and subsets

3D graphics is a big and complicated subject

- Beginning authors just want simple scenes
- Experienced authors want to use everything

Similar needs for browser software builders

- Small rapid download for simple web graphics
- Full-capability software for every possible technique

Challenge: how to consistently support both?

- Object-oriented decomposition for consistency
- Key design criteria for bottom-up X3D extensibility



These points are some of the original design challenges that faced X3D architects when evolving from the successes and lessons learned of VRML97.

# Profiles and components

Profiles are predefined collections of components

- Can augmented each by adding other components

Components are predefined collections of nodes

- Further defined by *level* of complexity
- Components match chapters in X3D specification

Authors define the expected complexity of scene by defining profile level in the X3D header

- Can also add optional components, if desired
- This tells the X3D browser what level of support is needed for run-time operation



Someday X3D browser software applications might themselves begin to componentize, enabling a light-weight initial download followed by run-time addition of further components as needed.

Each specification chapter includes a table at the end that lists the nodes and fields which are included for each component level.

This might sound a bit complicated, but is actually a helpful thing architecturally. Authors can simply choose the best profile, rarely needing to worry about the components or levels that make them up.

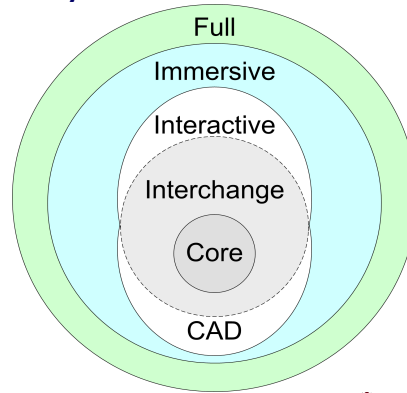
Further customization within a scene is always possible using `component` statements.

## Profiles cover common use cases

Profiles are a collection of components matching common levels of complexity

Profiles are X3D subsets

- Collection of X3D nodes for author's palette
- *Interchange* suitable for simple geometry conversion
- *Interactive* adds simple user interactivity (clicking etc.)
- Immersive matches VRML97
- Full profile includes all nodes



web|3D  
CONSORTIUM



This is the profiles and components “onion” diagram

# meta statements

meta statements provide information about the X3D scene

- Document metadata, not scene metadata

Information provided as name-value pairs

- Example:

```
<meta name='created' value='1 January 2008' />
```

This approach is thus very general

- Wide variety of metadata can be represented
- Matches same approach used by HTML for regular hypertext web pages



newScene.x3d includes a number of prompts for authors to fill in the proper metadata

<http://www.web3d.org/x3d/content/examples/newScene.x3d>

<http://www.web3d.org/x3d/content/examples/newScene.html>

A variety of metadata standards exist that specify the proper metadata terms to use. This allows consistent searchability among data files that follow the metadata norms.

<!-- Additional authoring resources for meta-tags:

<http://www.dublincore.org/documents/dcmi-terms>

<http://www.dublincore.org/documents/dces>

<http://www.w3.org/TR/html4/struct/global.html#h-7.4.4>

<http://vancouver-webpages.com/META>

<http://vancouver-webpages.com/META/about-mk-metas2.html>

Additional authoring resources for language codes:

<ftp://ftp.isi.edu/in-notes/bcp/bcp47.txt>

<http://www.loc.gov/standards/iso639-2/langhome.html>

<http://www.iana.org/numbers.html#L>

-->



## profile, component and meta statements, XML (.x3d) encoding syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.1//EN" "http://www.web3d.org/specifications/x3d-3.2.dtd">
<X3D version="3.2" profile="Immersive" xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance"
      xsd:noNamespaceSchemaLocation="http://www.web3d.org/specifications/x3d-3.2.xsd">
  <head>
    <component name='DIS' level='1'/>
    <component name='Geospatial' level='1'/>
    <component name='H-Anim' level='1'/>
    <component name='NURBS' level='4'/>
    <meta name='title' content='HeaderProfileComponentMetaExample.x3d'/>
  </head>
  <Scene>
    <!--Scene graph nodes are added here-->
  </Scene>
</X3D>
```

web|**3D**  
CONSORTIUM



## Profile, component and meta statements, ClassicVRML (.x3dv) encoding syntax

```
#X3D V3.2 utf8
PROFILE Immersive
# No HEAD statement is provided in ClassicVRML Encoding
COMPONENT DIS:1
COMPONENT Geospatial:1
COMPONENT H-Anim:1
COMPONENT NURBS:4
META "filename" "HeaderProfileComponentMetaExample.x3d"
# Scene graph nodes are added here
```



The book and slideset emphasize XML (.x3d) syntax over ClassicVRML (.x3dv) syntax but each may be used equivalently.

X3D-Edit is designed using the XML (.x3d) encoding, with ability to import/export ClassicVRML (.x3dv) and the Compressed Binary Encoding (.x3db). Native editing support for .x3dv (node coloration, popup menus, etc.) may be provided someday.

# newScene.x3d metadata prompts

```
<meta content='*enter FileNameWithNoAbbreviations.x3d here*' name='title'/>
<meta content='*enter description here, short-sentence summaries preferred*' name='description'/>
<meta content='*enter name of original author here*' name='creator'/>
<meta content='*if manually translating VRML-to-X3D, enter name of person translating here*' name='translator'/>
<meta content='*enter date of initial version here*' name='created'/>
<meta content='*enter date of translation here*' name='translated'/>
<meta content='*enter date of latest revision here*' name='modified'/>
<meta content='*enter version here, if any*' name='version'/>
<meta content='*enter reference citation or relative/online url here*' name='reference'/>
<meta content='*enter additional url/bibliographic reference information here*' name='reference'/>
<meta content='*enter reference resource here if required to support function, delivery, or coherence of content*' name='requires'/>
<meta content='*enter copyright information here* Example: Copyright (c) Web3D Consortium Inc. 2008' name='rights'/>
<meta content='*enter drawing filename/url here*' name='drawing'/>
<meta content='*enter image filename/url here*' name='image'/>
<meta content='*enter movie filename/url here*' name='MovingImage'/>
<meta content='*enter photo filename/url here*' name='photo'/>
<meta content='*enter subject keywords here*' name='subject'/>
<meta content='*enter permission statements or url here*' name='accessRights'/>
<meta content='*insert any known warnings, bugs or errors here*' name='warning'/>
<meta content='*enter online Uniform Resource Identifier (URI) or Uniform Resource Locator (URL) address for this file here*'
  name='identifier'/>
<meta content='X3D-Edit, https://savage.nps.edu/X3D-Edit' name='generator'/>
<meta content='../..//license.html' name='license'/>
```

# Field data types

X3D is a strongly typed language

- Each field in each node (i.e. each XML attribute) has a strictly defined data type
- Data types for boolean, integer, floating point

Types are either single or multiple-value

- Example: SFFloat, SFVec2f, SFVec3f, SFOrientation

Also have arrays for all types

SF = Single Field, MF = Multiple Field (array)

**Failure to match data types correctly is an error!**

- During schema validation, loading or at run time



Data types and accessType are covered in *X3D for Web Authors*, Chapter 7, Event Animation and Interpolation.

Data type and accessType information is available for each node in the X3D Tooltips and X3D Specification.

When speaking about data types, you can substitute “array of” for the “MF” prefix. Example: “MFColor is an array of Color values.”

## Field data types

Field-type names	Description	Example values
SFBool	Single-field boolean value	true or false (X3D syntax), TRUE or FALSE (ClassicVRML syntax)
MFBool	Multiple-field boolean array	true false false true (X3D syntax), [ TRUE FALSE FALSE TRUE ] (ClassicVRML syntax)
SFColor	Single-field color value, red-green-blue	0 0.5 1.0
MFColor	Multiple-field color array, red-green-blue	1 0 0, 0 1 0, 0 0 1
SFColorRGBA	Single-field color value, red-green-blue alpha (opacity)	0 0.5 1.0 0.75
MFColorRGBA	Multiple-field color array, red-green- blue alpha (opacity)	1 0 0 0.25, 0 1 0 0.5, 0 0 1 0.75 (red green blue, varying opacity)
SFInt32	Single-field 32-bit integer value	0
MFInt32	Multiple-field 32-bit integer array	1 2 3 4 5
SFFloat	Single-field single-precision floating- point value	1.0
MFFloat	Multiple-field single-precision floating- point array	−1 2.0 3.14159

*X3D for Web Authors*, Table 1.4, pp. 19-20.

## Field data types

Field-type names	Description	Example values
SFDouble	Single-field double-precision floating-point value	2.7128
MFDouble	Multiple-field double-precision array	-1 2.0 3.14159
SFImage	Single-field image value	Contains special pixel-encoding values, see Chapter 5 for details
MFImage	Multiple-field image value	Contains special pixel-encoding values, see Chapter 5 for details
SFNode	Single-field node	<Shape/> or Shape {space}
MFNode	Multiple-field node array of peers	<Shape/><Group/><Transform/>
SFRotation	Single-field rotation value using 3-tuple axis, radian angle form	0 1 0 1.57
MFRotation	Multiple-field rotation array	0 1 0 0, 0 1 0 1.57, 0 1 0 3.14
SFString	Single-field string value	"Hello world!"
MFString	Multiple-field string array	"EXAMINE" "FLY" "WALK" "ANY"
SFTime	Single-field time value	0
MFTime	Multiple-field time array	-1 0 1 567890

*X3D for Web Authors*, Table 1.4, pp. 19-20.

## Field data types

Field-type names	Description	Example values
SFVec2f/SFVec2d	Single-field 2-float/2-double vector value	0 1.5
MFVec2f/MFVec2d	Multiple-field 2-float/2-double vector array	1 0, 2 2, 3 4, 5 5
SFVec3f/SFVec3d	Single-field vector value of 3-float/ 3-double values	0 1.5 2
MFVec3f/MFVec3d	Multiple-field vector array of 3-float/ 3-double values	10 20 30, 4.4 –5.5 6.6

### ClassicVRML syntax notes

- TRUE and FALSE (rather than XML true and false)
- MF multiple-field array values are surrounded by square brackets, e.g. [ 10 20 30, 4.4 –5.5 6.6 ]
- No special XML escape characters such as **&amp;**;

*X3D for Web Authors*, Table 1.4, pp. 19-20.

**accessType: input, output, initialize**

accessType determines if field is data sender, receiver, or holder

- inputOnly: can only receive events
- outputOnly: can only send events
- initializeOnly: cannot send or receive
- inputOutput: can send, receive and be initialized

**Failure to match accessType correctly is an error!**

- Detected during authoring-tool checks, or run time



Data types and accessType are covered in *X3D for Web Authors*, Chapter 7, Event Animation and Interpolation.

Data type and accessType information is available for each node in the X3D Tooltips and X3D Specification.



## accessType naming conventions

The accessType names were changed when VRML97 was upgraded to X3D

- Functionality remains essentially unchanged

X3D specification entries for each node use yet another shorthand, as shown here

VRML97 Name	X3D Name	X3D Specification abbreviation
eventIn	inputOnly	[in]
eventOut	outputOnly	[out]
field	initializeOnly	[ ]
exposedField	inputOutput	[in,out]
VRML, Virtual reality modeling language; X3D, Extensible 3D.		

*X3D for Web Authors*, Table 1.6, p. 28.

# Abstract node types

X3D nodes also have strong typing

- Provides consistent field interfaces for similar nodes
- Object-oriented improvement over VRML97, which had several internal inconsistencies
- Better language design

Benefits include

- Allowed child-node content is consistent
- Simple-type field values have identical defaults
- Application programming interfaces more consistent
- Definitions are easier to remember and apply



# XML file encoding

The Extensible Markup Language (XML) is a plain-text format used by many Web languages

- Including Hypertext Markup Language (HTML)

XML is used to define other data-oriented languages

- Thus XML is not a language by itself, rather it is a language about languages, a *metalanguage*

XML has many benefits and is well-suited for X3D



# XML in 10 Points

<http://www.w3.org/XML/1999/XML-in-10-points>

XML is for structuring data

XML looks a bit like HTML

XML is text, but isn't meant to be read

XML is verbose by design

XML is a family of technologies

XML is new but not that new

XML leads HTML to XHTML

XML is modular

XML is basis for RDF and the Semantic Web

XML is license-free, platform-independent and well-supported

*XML in 10 Points* is a key reference for understanding the common underlying design principles underlying the great diversity of XML.

Only 4 pages long – essential reading.

web|3D  
CONSORTIUM



Bert Bos et al., “XML in 10 Points,: World Wide Web Consortium (W3C), created 1999, updated 2003. Available at <http://www.w3.org/XML/1999/XML-in-10-points>

# XML and X3D correspondence

Opening element	<Shape>
Singleton element, attribute="value"	<Sphere radius="10.0" solid="true"/>
Opening element	<Appearance>
Singleton element, attribute='value'	<ImageTexture url='earth-topo.png'/>
Closing element	</Appearance>
Closing element	</Shape>

Elements correspond to X3D nodes

Attributes correspond to X3D simple-type fields

Parent-child relationships define containerField

Validatable XML using X3D DTD, schema



XML documents have a tree structure that is a good match for the X3D scene graph.

# XML validation

XML validation applies XML rules to an XML document to confirm whether it is correct

- *Well formed XML*: legal header, matching open/close tags, proper attribute-value pairs, etc.
- *DTD (DOCTYPE) validation*: adds checks on legal element and attribute names, proper parent-child relationships, simple checks on attribute values
- *XML Schema validation*: also includes stricter datatype-aware checks on attribute values

XML validation finds problems before end users

- reducing garbage-in garbage-out (GIGO)



There are DTD and XML Schema definitions for X3D versions 3.1, 3.2 and 3.3

Additional quality-control checks are possible using special XSLT stylesheets and various X3D browsers themselves.

# ClassicVRML file encoding

The ClassicVRML file syntax is a direct, backwards-compatible extension of VRML97

- VRML version 2.0 became X3D version 3.0, 3.1 etc.
- No changes in syntax rules
- Some additional new nodes and slight naming differences to match specification improvements
- VRML97 content still works and is easily supported

XML, ClassicVRML and Compressed Binary encodings are functionally equivalent

- Governed by same X3D abstract specification



# Compressed binary encoding

Two types of compression for .x3db encoding

- XML-centric ISO Fast Infoset
- Geometry-centric for coplanar polygons, quantization of points, colors & normals, etc.

Java3D algorithms are default for geometry compression

- Royalty free for use with X3D
- Other uses – please contact Sun Microsystems

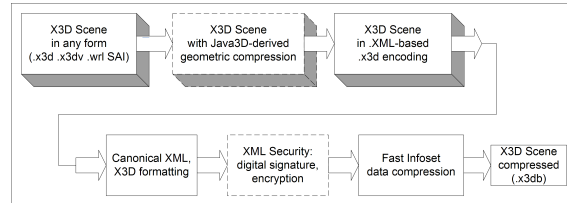
Alternate geometry compression is allowed

Sample implementation: Xj3D





## X3D compressed binary algorithm and XML Security



### X3D compressed binary uses Canonical X3D form

- Strict formatting rules so that files with identical format can be shown to match

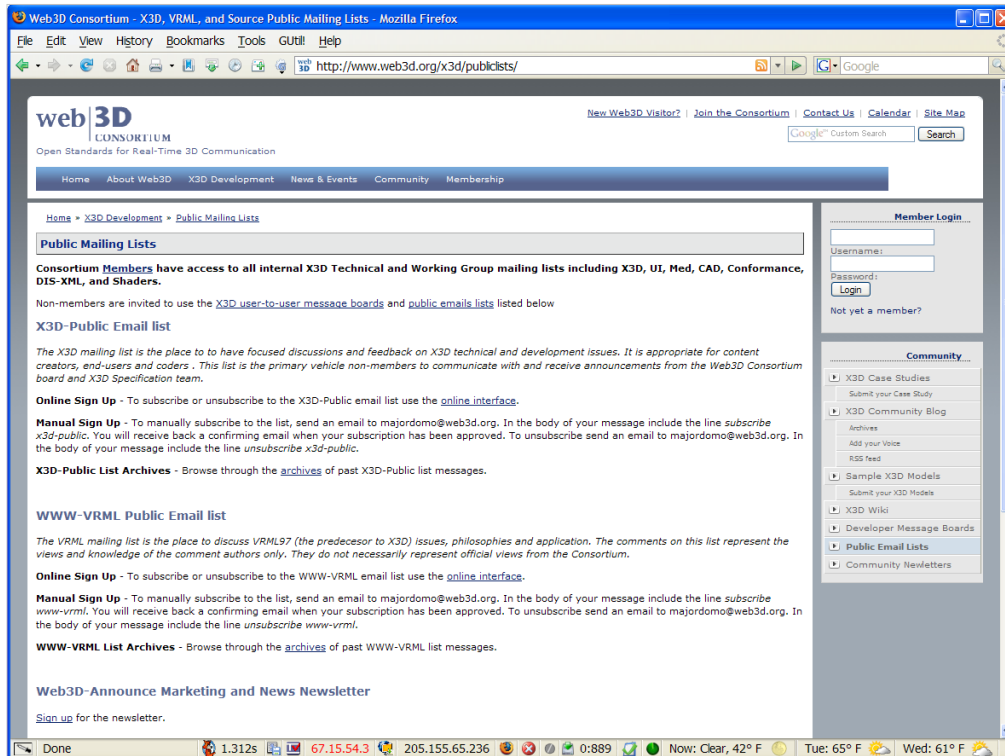
### Canonical form enables use of XML Security

- XML Encryption
- XML Digital Signature (for author authentication)

[back to Table of Contents](#)

## Additional Resources





There are two x3d mailing lists, plus more for other working groups

- [x3d-public@web3d.org](mailto:x3d-public@web3d.org)
- [x3d@web3d.org](mailto:x3d@web3d.org) private for Web3D members working on specification development

<http://www.web3d.org/x3d/publiclists>

# Annual Conferences

## SIGGRAPH

- <http://www.siggraph.org>

## Web3D Symposium

- <http://www.web3d.org/conferences/web3d2008>
- Web3D 2007 Tech Talk podcast

## Eurographics

- <http://www.eg.org>



# Web3D liaison organizations

## World Wide Web Consortium (W3C)

- Leading the Web to its Full Potential
- <http://www.w3.org>

## Open Geospatial Consortium

- Leading the development of standards for geospatial and location-based services.
- <http://www.opengeospatial.org>



[back to Table of Contents](#)

## Chapter Summary



## Chapter Summary

This technical overview chapter is a mile wide and a few meters deep. Key points include

- VRML historical background
- Web3D Consortium
- Browsers, X3D Specifications, scene graph
- Profiles + components, field and node data types
- XML encoding, ClassicVRML, Compressed binary

New students of X3D can refer to details later.

Get working on examples in the next chapters!



Students should have an [X3D plugin](#) installed in their [Web browser](#) by now, along with [X3D-Edit](#) or another editor.

[back to Table of Contents](#)

## References





# References 1

*X3D: Extensible 3D Graphics for Web Authors*  
by Don Brutzman and Leonard Daly, Morgan  
Kaufmann Publishers, April 2007, 468 pages.

- Chapter 3, Grouping Nodes
- <http://x3dGraphics.com>
- <http://x3dgraphics.com/examples/X3dForWebAuthors>

## X3D Examples Help

- <http://www.web3d.org/x3d/content/examples/help.html>



## References 2

### X3D Scene Authoring Hints

- <http://x3dgraphics.com/examples/X3dSceneAuthoringHints.html>

### X3D Graphics Specification

- <http://www.web3d.org/x3d/specifications>
- Also available as help pages within X3D-Edit



## References 3

*Computer Graphics, Principles and Practice*, by James D. Foley, Andries van Dam, Stephen K. Feiner and John F. Hughes, Addison-Wesley, 2nd Edition, 1997.

Bert Bos et al., "XML in 10 Points," World Wide Web Consortium (W3C), created 1999, updated 2003. Available at <http://www.w3.org/XML/1999/XML-in-10-points>



# Book testimonials 1

There will be no problem understanding these concise, clear, comprehensible background concepts for readers new to Extensible 3D (X3D). There are many notes and examples that compare X3D to Virtual Reality Modeling Language (VRML) features. Don Brutzman and Leonard Daly clearly and thoroughly illustrate each logical concept and feature of X3D with diagrams, tables, code snippets, screenshots of 3D objects/environments, and example scenes, while making use of the very latest specifications and implementations. Their approach contributes greatly to an easy and in-depth understanding of the X3D language. This book is the ultimate introductory guide to X3D!

—Dr. Vladimir Geroimenko, University of Plymouth,  
School of Computing Communications and Electronics, Plymouth, UK



## Book testimonials 2

This book is required reading for anybody interested in Web3D. The authors are well known and respected in the X3D community as pioneers. Their writing style is concise and engaging, set at an appropriate level to encourage understanding, and uses the concepts being introduced. Their “Hints and warnings” sections provide added value above what is available from X3D specification documents. Hard to achieve in a reference manual!

—Professor Nigel W. John, School of Computer Science,  
University of Wales, Bangor; Chair of Web3D 2005 Symposium



## Book testimonials 3

How many times have we heard “The ISO specification is hard to read, do you have something more approachable?” This book is the answer. It provides a detailed explanation of each node in the Immersive profile and gives many reusable examples. After reading this book you’ll be well prepared to develop your own X3D content.

—Alan Hudson, President Web3D Consortium, Yumetech Inc.

This is a much-needed book about the X3D standard and X3D content development. The book follows the structure of the X3D standard specifications which helps readers understand and apply the X3D standard. It can also be used as a reference material in virtual reality and graphics-related courses.

—Professor Denis Gracanin, Virginia Polytechnic Institute & State University, Chair Web3D 2006 Symposium



# Contact

**Don Brutzman**

*[brutzman@nps.edu](mailto:brutzman@nps.edu)*

*<http://web.nps.navy.mil/~brutzman>*

Code USW/Br, Naval Postgraduate School  
Monterey California 93943-5000 USA

1.831.656.2149 voice

1.831.656.7599 fax



# Open-source license

Copyright (c) 1995-2008 held by the author(s). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the Naval Postgraduate School (NPS) Modeling Virtual Environments and Simulation (MOVES) Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.