

# Chapter 2

## Interactive 3D Content Standards

Wojciech Cellary and Krzysztof Walczak

**Abstract** Development of 3D interactive network applications requires standards for representing 3D content as well as metadata standards for describing the content. In this chapter, selected standards for content representation and content description are presented. In particular, the VRML, X3D, and MPEG-4 ISO/IEC standards are presented. Other standards such as U3D, COLLADA, and 3D XML are also discussed and compared. Metadata standards suitable for describing interactive 3D content are also presented.

### 2.1 Content Representation Standards

A number of standards have been developed for platform-independent representation of 3D/VR content permitting exchange of content between applications and its distribution through the network. Content representation standards differ in their capabilities of describing content features and encoding methods, making them more suitable either for exchange or for publishing of content. In this section, selected standards suitable for content publishing are presented.

The most versatile content publishing standards are VRML [27, 28], X3D [37], and MPEG-4 [29, 41], all approved by ISO/IEC. These standards enable publishing or broadcasting synthetic interactive multimedia 3D content and accessing the content on various platforms in different architectural setups. The U3D ECMA standard [51] enables embedding and presentation of 3D models within PDF documents. Content exchange formats, which also enable content visualization in Web browsers, include COLLADA [5, 14] and 3D XML [17]. All these standards are further described in the following sections.

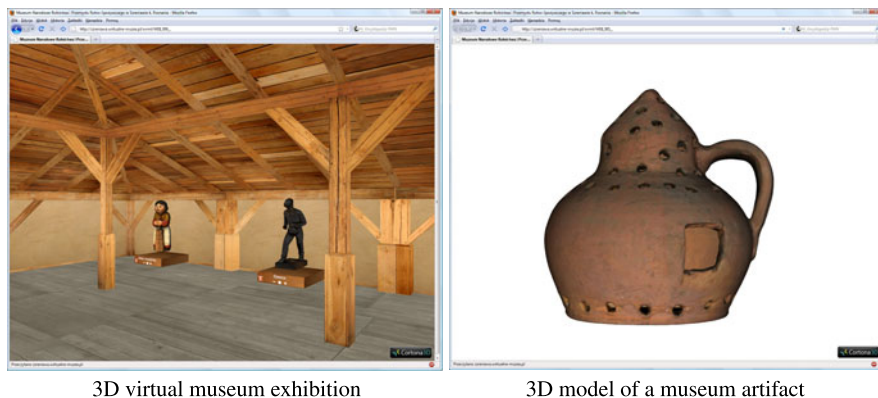
#### 2.1.1 Virtual Reality Modeling Language

The *Virtual Reality Modeling Language (VRML)* is a textual file format for representing and publishing interactive 3D multimedia content [4, 11, 27, 28]. VRML is

---

W. Cellary · K. Walczak

Department of Information Technology, Poznań University of Economics, Poznań, Poland



**Fig. 2.1** VRML scenes displayed in a Web browser

capable of representing both static and animated 3D and multimedia objects with hyperlinks to other media such as text, sounds, movies, and images.

VRML browsers, as well as authoring tools enabling creation of VRML files, are available for different platforms. Currently, the most popular browsers include *ParallelGraphics Cortona3D* [15], *Bitmanagement BS Contact* [9], open-source *Xj3D* [65], and *instantreality* [24]. Figure 2.1 shows two examples of 3D VRML models displayed in a Web browser equipped with a VRML browser plug-in (3D models courtesy of the National Museum of Agriculture and Agricultural Food Industry in Szreniawa [56]).

VRML has been developed by the Web3D Consortium [60]. The first release of the VRML specification (VRML 1.0) [7] was created in 1994 based on the *Open Inventor* data exchange format [62, 63]. That version of the specification allowed creation of static only virtual scenes. The second release—VRML 2.0—added support for animation, interaction, and scripting. In December 1997, VRML 2.0 with small corrections was formally released as an International Standard ISO/IEC 14772:1997 [27]. This version is commonly known as VRML97. The specification consists of two parts: *Part 1* (ISO/IEC 14772-1) defines the base functionality and text encoding (UTF-8) for VRML. *Part 2* (ISO/IEC 14772-2) defines the base functionality and language bindings for the *VRML External Authoring Interface (EAI)*. In 2003, Amendment 1 to the specification (ISO/IEC14772-1:1997/Amd. 1:2003) was formally approved [28]. The amendment adds modifications to allow better interoperability among VRML implementations as well as support for geographical objects (*GeoVRML*) and *NURBS* (Non-Uniform Rational B-Spline) nodes. Currently, VRML has been officially superseded by X3D (Sect. 2.1.2), however, the original VRML97 specification is still widely used among developers.

VRML has been designed for use on the Internet, intranets, and local client systems. It was intended to be a universal publishing and interchange format for 3D graphics and multimedia. VRML is in some sense analogous to HTML—it is a multi-platform language for publishing 3D content on the World Wide Web. It supports also the notion of hyperlinks. Entities in a VRML virtual scene can be con-

nected via hyperlinks to other scenes and other media such as text, sounds, movies, and images.

VRML describes multimedia content in an abstract way, without defining any physical devices or other implementation-dependent concepts (e.g., screen resolution or input devices). Each VRML file describes a single virtual scene. The scene may be the whole “virtual world,” a part of it, a single virtual object, or a part of a virtual object. One virtual scene may play different roles in different contexts. Each VRML file establishes a coordinate space for all objects defined and included in this file, defines and arranges in this space a set of 3D and multimedia objects, and can specify hyperlinks to other VRML or non-VRML Web resources.

VRML files describe contents of 3D scenes and 3D objects using a hierarchical structure called a *scene graph*. Elements of the scene graph are called nodes. VRML defines a multitude of different types of nodes. These types include geometry primitives such as *box*, *sphere*, *cone*, *indexed-face-set*, and *text*, appearance properties such as *material*, *image texture*, and *movie texture*, sounds and sound properties, and various types of grouping nodes such as *group*, *transformation*, *inline*, and *level-of-detail*.

Properties of nodes are defined in their fields. VRML specification provides a list of permitted field types including both single- and multi-value types. Basic field types include *integers*, *floats*, *Boolean values*, *strings*, *time stamps*, *color values*, *images*, different types of *vectors*, and *nodes*.

VRML uses a *dataflow* based event-passing mechanism for communication between nodes in the scene graph. Each type of nodes defines names and types of events that instances of this type may generate or receive. Special *ROUTE* statements are used to create paths for events between event-generators and event-receivers. Nodes can send events upon event reception thus leading to event cascades.

There is a special group of nodes in VRML called *sensors*. They are the basic mechanism allowing users’ interaction and animation in virtual scenes. There are several different types of sensors in VRML. These include a *time sensor*, a *proximity sensor*, a *visibility sensor*, and a variety of pointing device sensors—*anchor*, *cylinder sensor*, *plane sensor*, *sphere sensor*, and *touch sensor*. All sensors, except the time sensor, generate events in response to some user actions. They can be connected via *ROUTE* statements to other nodes in the scene to implement interactivity. The time sensor is a special node that generates events as the time passes. It is utilized in all kinds of animations in virtual scenes.

VRML provides a possibility of programming scene behaviors by the use of a special *Script* node. Script nodes can be inserted between event generators and event receivers. A script is a program executed every time an input event is received. The program can generate output events during its execution.

Smooth animations in VRML are achieved by the use of special *interpolator* nodes. Interpolator nodes behave like scripts but are built-in in each VRML-compliant browser. Interpolators perform simple animation calculations and are usually combined in a scene with a time sensor and some other nodes to implement movement in the scene.

**List. 2.1** Example of a simple VRML scene

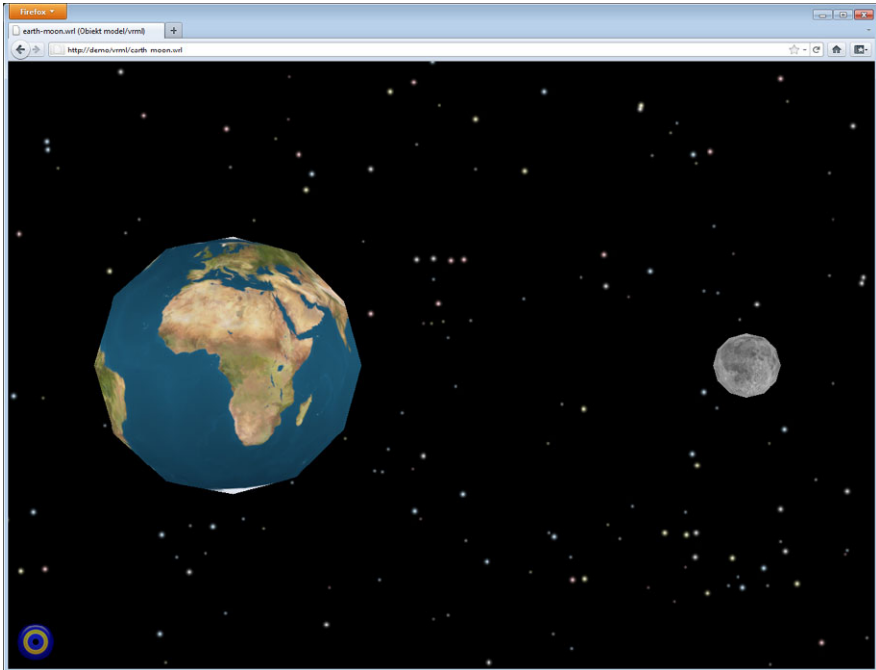
```
#VRML V2.0 utf8
Background {
    frontUrl "stars.jpg"
}
Transform {
    translation -2 0 0
    children [
        Shape {
            geometry Sphere { radius 1 }
            appearance Appearance {
                texture ImageTexture { url "earth.jpg" }
            }
        }
    ]
}
Transform {
    translation 2 0 0
    children [
        Shape {
            geometry Sphere { radius 0.25 }
            appearance Appearance {
                texture ImageTexture { url "moon.jpg" }
            }
        }
    ]
}
```

An example of a VRML scene code is presented in List. 2.1, while the scene rendered in a VRML browser is presented in Fig. 2.2.

### 2.1.2 Extensible 3D

*X3D—Extensible 3D*—is the successor to VRML (Sect. 2.1.1) also developed by the *Web3D Consortium* [60]. X3D has been designed to keep backward compatibility with VRML97 while providing more advanced functionality, new encoding formats, componentization, and extensibility.

X3D provides some functional extensions such as *Humanoid Animation (H-Anim)*, *Distributed Interactive Simulation (DIS)*, *CAD geometry*, *programmable shaders*, *scene layering*, *rigid body physics*, and *particle systems*. X3D also provides new formats for encoding virtual scenes. In addition to the VRML97 encoding, *XML encoding* and *binary encoding* are allowed in X3D. The standard also offers enhanced application programming interfaces (APIs).



**Fig. 2.2** VRML virtual scene from List. 2.1 rendered in a Web browser with BS Contact plug-in

The X3D specification is modular. It defines a set of componentized elements that can be tailored for use in various applications or on various platforms. Modular specification simplifies creation of browsers and authoring tools for X3D.

The modular structure of the X3D standard allows definition of *profiles* offering different levels of functionality for different purposes. Specification of a profile consists of a list of required X3D components and their support levels. Different X3D systems can conform to different X3D profiles depending on their particular architectural or application requirements.

The main X3D profiles are:

- *Core profile*—defining the absolute minimal file support required by X3D.
- *Interchange profile*—the basic profile for exchanging X3D content—geometry and animations—between applications. It supports geometry, texturing, basic lighting, and animation. There is no runtime model for rendering and interaction, making it easy to use and integrate into any application.
- *Interactive profile*—enabling basic interaction with a 3D environment by adding various sensor nodes for user navigation and interaction (e.g., *PlaneSensor*, *TouchSensor*, etc.), enhanced timing, and additional lighting (*Spotlight*, *Point-Light*).
- *Immersive profile*—enabling implementation of immersive virtual worlds with complete 3D graphics and interaction support, including audio, collision, fog,

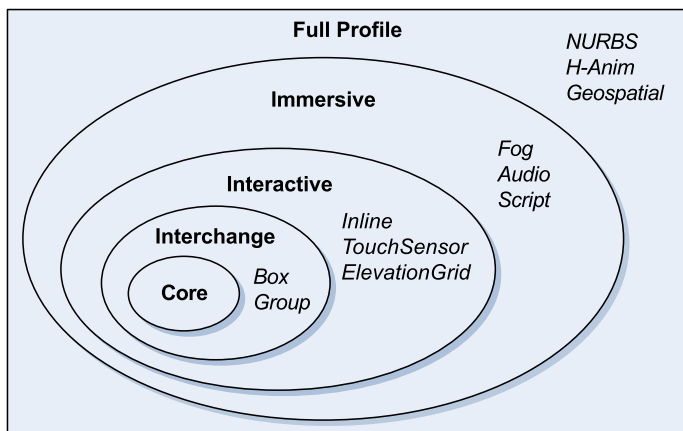


Fig. 2.3 Relationships between main X3D profiles

and scripting. The immersive profile corresponds to *VRML97 base profile*, but implemented in the X3D architectural framework.

- *Full profile*—enabling the use of all nodes defined in X3D, including *NURBS*, *H-Anim* and *Geospatial* components.

Additional profiles defined in the X3D standard are:

- *MPEG-4 Interactive profile*—providing basic interoperability with the MPEG-4 standard (Sect. 2.1.3) targeting broadcast, handheld devices, and mobile phones.
- *CAD Interchange profile*—enabling translation of CAD data for use by downstream applications, while appropriately supporting geometry and appearance capabilities of CAD systems.

Relationships between the main X3D profiles are graphically presented in Fig. 2.3. In Table 2.1 below, the ISO standardization status of VRML and X3D is presented [64].

In List. 2.2, X3D XML encoding of the virtual scene from List. 2.1 is presented. Figure 2.4 shows the X3D scene displayed in the *instantreality* browser.

### 2.1.3 MPEG-4

*MPEG-4* is an ISO/IEC standard developed by the *Moving Picture Experts Group* (MPEG) [55], a working group of the subcommittee *SC29 Coding of audio, picture, multimedia and hypermedia information* of the *Joint Technical Committee 1* of the *International Organization for Standardization (ISO)* [26] and the *International Electrotechnical Commission (IEC)* [25]. The MPEG group has also developed the *MPEG-1* and *MPEG-2* [12] as well as *MPEG-7* [36, 42] (Sect. 2.2.3) and *MPEG-21* [8, 38] standards.

MPEG-4 offers a comprehensive set of tools for delivery of different kinds of multimedia content [47]. The MPEG-4 standard provides standardized methods of

**Table 2.1** ISO standardization status of VRML/X3D (as of July 2011)

ISO/IEC name	Common name	Status	Updated
ISO/IEC PDAM1 19775-1:2008	X3D Architecture and base components V3 (Change Document)	PDAM	July 2011
ISO/IEC 19775-1.2:2008	X3D Architecture and base components Edition 2	IS	July 2008
ISO/IEC 19775-2.2:2010	X3D Scene access interface Edition 2	IS	Jan 2011
ISO/IEC 19776-1.2:2009	X3D encodings: XML encoding Edition 2	IS	Oct 2009
ISO/IEC 19776-2.2:2008	X3D encodings: Classic VRML encoding Edition 2	IS	Oct 2008
ISO/IEC 19776-3:2007	X3D encodings: Compressed binary encoding Edition 1	IS	Sep 2007
ISO/IEC FDIS 19776-3.2:2011	X3D encodings: Compressed binary encoding Edition 2	FDIS	Jan 2011
ISO/IEC 19777-1:2006	X3D language bindings: ECMAScript	IS	May 2006
ISO/IEC 19777-2:2006	X3D language bindings: Java	IS	May 2006
ISO/IEC 19774:2006	Humanoid Animation	IS	June 2006
ISO/IEC 14772:1997	Virtual Reality Modeling Language (VRML97)	IS	Dec 2003
ISO/IEC 14772- 1:1997/Amd. 1:2002	VRML97 Amendment 1	IS	Dec 2003

[41]:

- representing units of aural, visual or audiovisual content, called *media objects*,
- composing these media objects into audiovisual scenes,
- multiplexing and synchronizing data associated with media objects,
- interaction with the audiovisual scene generated at the receiver side.

Examples of MPEG-4 application types, include [57, 59]:

- digital television (broadcasted and IP-based),
- mobile communication, entertainment and portable gaming,
- packaged media distribution,
- video conferencing,
- home networking, video recorders and cameras, surveillance systems,
- satellite radio.

## Representation of Media Objects

MPEG-4 standardizes a number of primitive types of media objects, which can represent both *natural* and *synthetic content* types. These media objects can be either

**List. 2.2** Code of the VRML scene from List. 2.1 encoded in X3D format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
"http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D version="3.0" profile="Immersive">
  <Scene>
    <Background frontUrl="stars.jpg"/>
    <Transform translation="-2 0 0">
      <Shape>
        <Sphere radius="1"/>
        <Appearance>
          <ImageTexture url="earth.jpg"/>
        </Appearance>
      </Shape>
    </Transform>
    <Transform translation="2 0 0">
      <Shape>
        <Sphere radius="0.25"/>
        <Appearance>
          <ImageTexture url="moon.jpg"/>
        </Appearance>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

two- or three-dimensional. Primitive media objects types include:

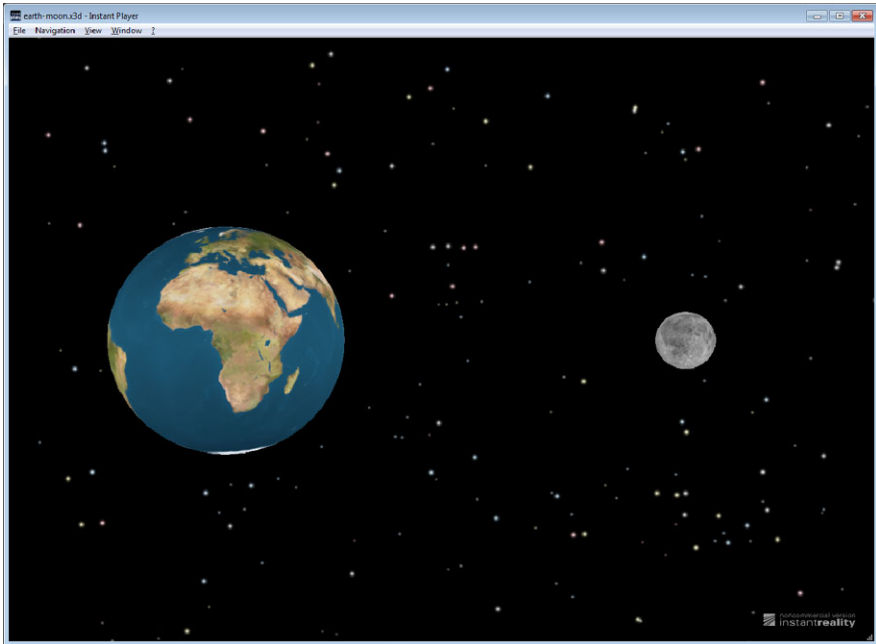
- still images,
- video objects,
- audio objects,
- text and graphics,
- synthetic objects,
- synthetic sound.

Each coded media object consists of descriptive information that enables proper handling of the object in an audiovisual scene and streaming data associated with the object if necessary. Each media object can be represented independently of its surroundings or background.

### Composition of Audiovisual Scenes

Audiovisual scenes in MPEG-4 are organized in a form of trees of media objects (similarly as in VRML and X3D). Leaves of the trees correspond to primitive media objects, while sub-trees correspond to compound media objects. For example, a





**Fig. 2.4** The X3D scene from List. 2.2 displayed in the *instantreality* browser

visual object can be tied together with a corresponding sound to form a new compound media object. Hierarchical grouping allows authors to construct complex audiovisual scenes and enables users to manipulate meaningful sets of objects in the receiver.

The MPEG-4 standard enables positioning media objects in the coordinate system, applying transformations to media objects to change their geometrical or aural appearance, grouping primitive media objects in order to form complex media objects, applying streaming data to media objects in order to modify their visual or aural appearance, and changing the user's viewing and listening points in the scene.

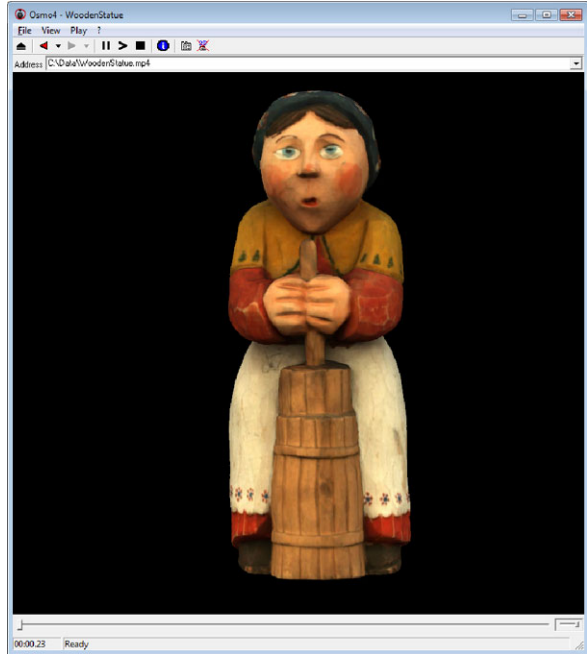
The scene representation in MPEG-4 standard is based on the scene representation of VRML/X3D and extended to cover full MPEG-4 functionality. The representation uses two encodings: binary *BIFS* and textual *BIFS-Text*. Although *BIFS-Text* is not part of the standard, it is commonly used.

An example of an MPEG-4 scene displayed in the *Osmo4* player [22] is presented in Fig. 2.5 (3D model courtesy of [56]).

### **Interaction with Audiovisual Scenes**

A user observes an MPEG-4 scene in the form that has been designed by its author. However, the author can enable the user to interact with the MPEG-4 scene. Depending on the degree of freedom allowed by the author, the user may have different possibilities of interaction with the audiovisual scene. Similarly as in VRML/X3D,

**Fig. 2.5** Example of a 3D MPEG-4 scene displayed in the Osmo4 player



the following operations may be allowed:

- change the viewing/listening point in the scene (navigate),
- drag/rotate/resize objects in the scene (manipulate),
- trigger events by clicking, pointing at, or approaching a specific object.

More complex kinds of behavior can be programmed by the use of scripting mechanisms based on *Java* (through *MPEG-J* [52]) or *ECMAScript*.

### Extensible MPEG-4 Textual Format

The *Extensible MPEG-4 Textual format (XMT)* is a framework for representing MPEG-4 scenes using a textual XML syntax. XMT allows content authors to exchange their content with other authors, tools or service providers, and facilitates interoperability with both the X3D and the *Synchronized Multimedia Integration Language (SMIL)* by the *W3C Consortium* [10, 53].

The XMT framework consists of two levels of textual syntax and semantics: *XMT-A format* and *XMT- $\Omega$  format*. XMT-A is an XML-based version of MPEG-4 content which contains a subset of X3D and extends it to represent MPEG-4 specific features. XMT-A provides one-to-one mapping between the textual and binary formats. In List. 2.3, the overall structure of XMT-A representation of an MPEG-4 scene is presented. XMT- $\Omega$  is a high-level abstraction of MPEG-4 elements. The goal of XMT- $\Omega$  is to facilitate content interchange and provide interoperability with the W3C SMIL language.

**List. 2.3** Structure of XMT-A representation of an MPEG-4 scene

```

<XMT-A xmlns='...'>
  <Header>
    <meta>
    </meta>
    <InitialObjectDescriptor/>
  </Header>
  <Body>
    <Replace>
      <Scene>
        <!--The scene contents-->
        ...
      </Scene>
    </Replace>
  </Body>
</XMT-A>

```

**Structure of the MPEG-4 Standard**

MPEG-4 consists of separate but closely interrelated parts that can be implemented individually or combined with other parts. The basis is formed by Part 1—*Systems*, Part 2—*Visual* and Part 3—*Audio*. Part 4—*Conformance* defines implementation testing, Part 5—*Reference Software* provides reference implementations, while Part 6—*DMIF (Delivery Multimedia Integration Framework)* defines an interface between MPEG-4 applications and network/storage. All parts of the MPEG-4 standard available or being developed at the time of writing this book are listed in Table 2.2. Relevant parts are highlighted and then described shortly.

Part 1—*Systems*—specifies system-level functionalities for communication of interactive audio-visual scenes, including a terminal model, representation of metadata of elementary streams, representation of object content information, interface to IP management and protection systems as well as representation of synchronization information and representation of individual elementary streams in a single stream [29].

Part 2—*Visual*—provides elements related to the encoded representation of visual information, including: specification of video coding tools, mapping of still textures into visual scenes, human face and body animation based on face/body models, and animation of 2D deformable meshes [30].

Part 11—*Scene description and application engine*—specifies the coded representation of interactive audio-visual scenes and applications. In particular, it defines the following elements: spatio-temporal positioning of audio-visual objects as well as their behavior in response to interaction, representation of synthetic two-dimensional (2D) and three-dimensional (3D) objects, the *Extensible MPEG-4 Textual format (XMT)*, and description of an application engine [31].

Part 16—*Animation Framework eXtension (AFX)*—specifies extensions for representing 3D graphics content. It provides higher-level synthetic objects for speci-

**Table 2.2** Parts of the MPEG-4 standard—ISO/IEC 14496 (as of July 2011)

---

<b>1</b>	<b>Systems</b>
<b>2</b>	<b>Visual</b>
3	Audio
4	Conformance testing
5	Reference software
6	Delivery Multimedia Integration Framework (DMIF)
7	Optimized reference software for coding of audio-visual objects
8	Carriage of ISO/IEC 14496 contents over IP networks
9	Reference hardware description
10	Advanced Video Coding (AVC)
<b>11</b>	<b>Scene description and application engine</b>
12	ISO base media file format
13	Intellectual Property Management and Protection (IPMP) extensions
14	MP4 file format
15	Advanced Video Coding (AVC) file format
<b>16</b>	<b>Animation Framework eXtension (AFX)</b>
17	Streaming text format
18	Font compression and streaming
19	Synthesized texture stream
<b>20</b>	<b>Lightweight Application Scene Representation (LAsER) and Simple Aggregation Format (SAF)</b>
<b>21</b>	<b>MPEG-J Graphics Framework eXtensions (GFX)</b>
22	Open Font Format
23	Symbolic Music Representation
24	Audio and systems interaction
<b>25</b>	<b>3D Graphics Compression Model</b>
26	Audio conformance
27	3D Graphics conformance
28	Composite font representation

---

fyng geometry, texture, and animation as well as dedicated compression methods. AFX also specifies a backchannel for progressive streaming of view-dependent information. In addition, the standard defines profiles for using MPEG-4 3D graphics tools in applications [32].

Part 20—*Lightweight Application Scene Representation (LAsER) and Simple Aggregation Format (SAF)*—defines a scene representation format (LAsER) and an aggregation format (SAF) suitable for representing and delivering rich-media services to resource-constrained devices such as mobile phones. A rich media service is a dynamic, interactive collection of multimedia data such as audio, video, graphics, and text. Example services range from simple movies enriched with graphic

overlays and interactivity to complex services with fluid interaction and a variety of media types [33].

Part 21—*MPEG-J Graphics Framework eXtensions (GFX)*—describes a light-weight programmatic environment for interactive multimedia applications on devices with limited resources, e.g., mobile phones. GFX offer a framework that combines a subset of the MPEG Java application environment (MPEG-J) with a Java API for accessing 3D renderers, and other Java APIs. The framework enables creation of applications that combine audio and video streams with 3D graphics rendering and user interaction. By reusing existing mobile APIs, most Java applications designed for mobile devices may be ported to a system using GFX. A GFX implementation is a thin layer of classes and interfaces over standard Java and MPEG-J APIs [34].

Part 25—*3D Graphics Compression Model*—defines methods of applying 3D graphics compression tools defined in MPEG-4 to potentially any XML-based representation format for scene graphs and graphics primitives. The model has been implemented for XMT, COLLADA (Sect. 2.1.5), and X3D (Sect. 2.1.2) [35, 49].

### 2.1.4 Universal 3D

*Universal 3D (U3D)* is a file format developed by *Intel* and the *3D Industry Forum (3DIF)* [1] including companies such as *Intel*, *Boeing*, *Hewlett-Packard*, *Adobe Systems*, *Bentley Systems*, *Right Hemisphere*, and others. The format has been standardized by *ECMA International* as ECMA-363 [51].

The U3D file format has been designed to enable repurposing and presentation of 3D CAD models. Examples of applications of the U3D file format include:

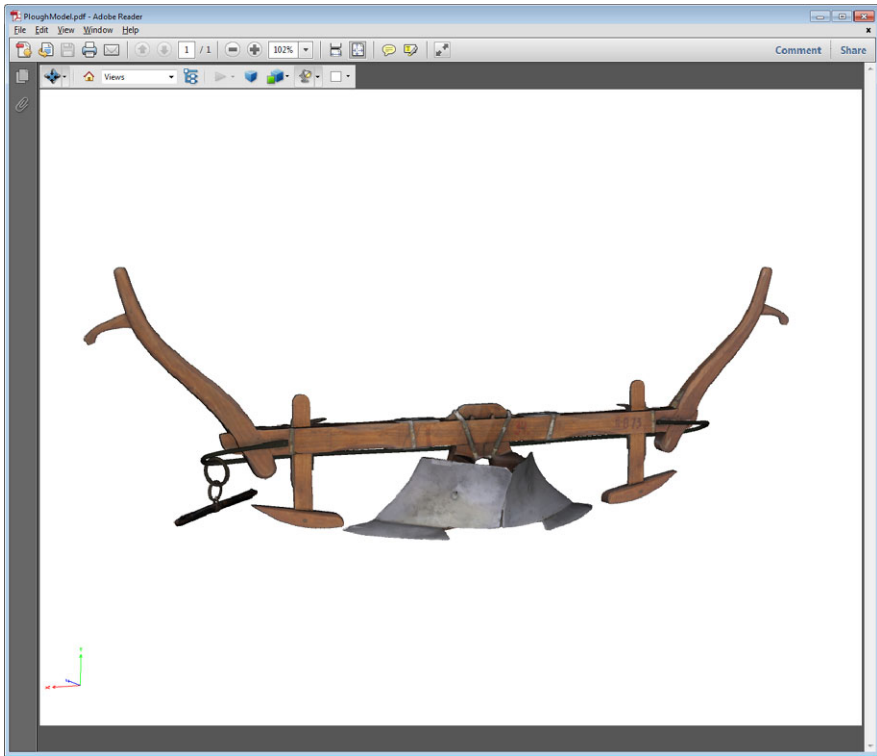
- development of training tools based on interactive simulations created from 3D CAD models;
- electronic owner's manuals providing interactive guides for maintaining and repairing products;
- online catalogues equipped with interactive 3D models created during product development.

The most important features of the U3D file format include possibility of runtime modification of geometry, continuous-level-of-detail, domain-specific compression, progressive data streaming and playback, free-form surfaces, key-frame and bones-based animation, and extensibility.

The U3D format is supported by PDF; 3D objects in U3D format can be inserted into PDF documents and interactively visualized by *Adobe Reader* (Fig. 2.6) [2] (3D model courtesy of [56]).

### 2.1.5 COLLADA

*COLLADA (COLLABorative Design Activity)* is an intermediate format designed for content interchange between 3D authoring applications. The work on COLLADA



**Fig. 2.6** U3D model embedded in a PDF document and displayed in *Adobe Reader X*

has been started in 2003 by *Sony Computer Entertainment* supported by a growing group of other companies. In 2005, *Khronos Group* [54] accepted COLLADA as industry standard [13]. Similarly as X3D, COLLADA is a royalty-free open standard [6].

COLLADA targets mainly the game industry, focusing on exchanging content and assets between diverse authoring tools, including *DCC (Digital Content Creation)* tools and *Conditioning* tools, and for archiving rich content. COLLADA enables free exchange of asset data, and therefore enables combining multiple tools into an integrated and flexible authoring pipeline.

COLLADA specifies an open standard XML schema for describing content. COLLADA documents are XML files, usually with a *\*.dae* (digital asset exchange) filename extension. A COLLADA XML file contains a root element `<COLLADA>`, which must contain an `<asset>` element and may contain a number of various library elements, a `<scene>` element, and a number of `<extra>` elements. An example of a COLLADA file is presented in List. 2.4.

Although COLLADA is not intended to be a content publishing format, there are browsers capable of displaying COLLADA content. An example COLLADA scene rendered in a Web browser equipped with the *BS Contact* plug-in is presented

**List. 2.4** Example of a COLLADA file (shortened)

```

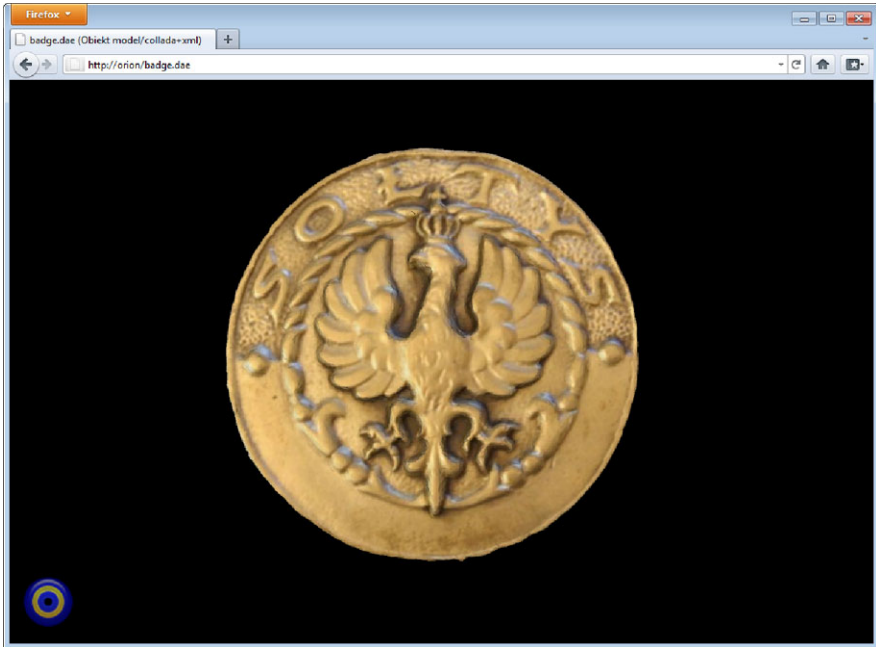
<?xml version="1.0" encoding="utf-8"?>
<COLLADA xmlns="http://www.collada.org/2005/11/COLLADASchema"
  version="1.4.1">
  <asset>
    <contributor>
      <author>walczak</author>
    </contributor>
    <created>2011-07-22T00:34:10</created>
  </asset>
  <library_materials>...</library_materials>
  <library_geometries>
    <geometry id="geom-BADGE" name="1_FACES001">
      <mesh>
        <source id="geom-BADGE-positions">
          <float_array id="geom1" count="212706">-3.30721
            79.6279... </float_array>
        </source>
        <triangles material="VMt1001" count="23634">
          <p>0 0 0 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 ... </p>
        </triangles>
      </mesh>
    </geometry>
  </library_geometries>
  <library_cameras>...</library_cameras>
  <library_lights>...</library_lights>
  <library_images>...</library_images>
  <library_visual_scenes>
    <visual_scene id="Scene">
      <node id="node-BADGE" name="BADGE">
        <instance_geometry url="#geom-BADGE">
          ...
        </instance_geometry>
      </node>
    </visual_scene>
  </library_visual_scenes>
  <scene>
    <instance_visual_scene url="#Scene"/>
  </scene>
</COLLADA>

```

in Fig. 2.7 (3D model courtesy of [56]). COLLADA models can be converted into VRML and X3D formats [48].

### 2.1.6 3D XML

*3D XML* is a proprietary XML-based 3D content representation format developed by *Dassault Systèmes* [17]. The 3D XML format is used in the whole suite of Dassault



**Fig. 2.7** COLLADA scene displayed in a Web browser with *BS Contact* plug-in

Systèmes' tools including *CATIA*, *DELMIA*, *ENOVIA*, *SIMULIA*, *SolidWorks*, and *Virtools*. A free 3D XML player—*3DVIA Player*—is available both as a plug-in for Web browsers and as a standalone application [16].

A 3D XML package is made up of a collection of separate files containing data such as product structure, geometric representations, images, materials, and industry-specific data. All files within an archive are compressed using ZIP compression. A special *Manifest.xml* file identifies the starting point in the archive. The structure of all 3D XML files is described using XML Schema [58], although the standard can also include binary representation of geometry.

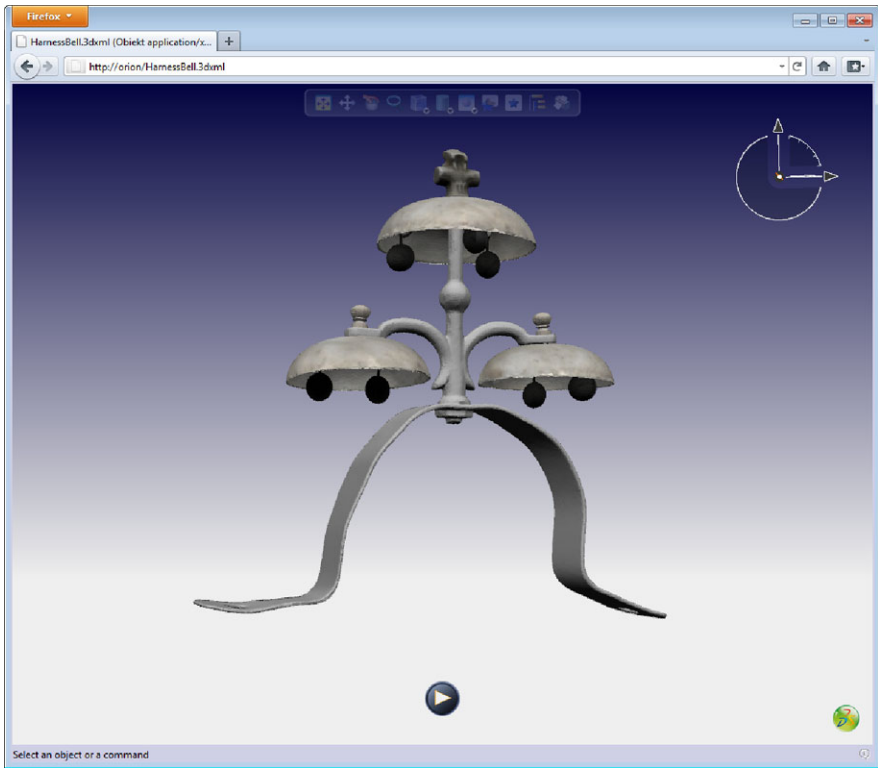
3D XML format has been primarily developed to support *PLM (Product Life-cycle Management)* tools, and consequently it is limited to the description of 3D designs, without support for interaction and behavior.

An example of a 3D XML model of a museum artifact rendered in a Web browser equipped with the *3DVIA* plug-in is presented in Fig. 2.8 (3D model courtesy of [56]).

### **2.1.7 Comparison of 3D Content Standards**

In previous sections, popular 3D content standards have been described. These standards have been designed for different purposes and—consequently—have different





**Fig. 2.8** 3D XML model displayed in a Web browser with the 3DVIA plug-in

**Table 2.3** Characteristics of 3D content standards

Standard	Main purpose	Animation	Interaction	Media
VRML	Publishing	Yes	Yes	Images, Sounds, Movies, Texts
X3D	Publishing	Yes	Yes	Images, Sounds, Movies, Texts
MPEG-4	Publishing	Yes	Yes	Images, Sounds, Movies, Texts
U3D	Publishing	Yes	External	Images
COLLADA	Exchange	Yes	No	Images
3D XML	Exchange Publishing	No	No	Images

abilities. Some of them are meant as mainly content exchange formats, while others are more suitable for publishing. These standards also differ in their capabilities to handle different types of media content such as sounds, movies, and graphics. In Table 2.3, main characteristics of the described 3D content standards are summarized.

## 2.2 Metadata for Describing Multimedia Content

Metadata can be used to describe contents and characteristics of data resources. Metadata descriptions permit to more efficiently organize, search, access, and interpret content, and therefore are critical to facilitate content reuse. Different types of metadata may be used for different types of content and for different application areas.

Numerous metadata standards have been developed to enable description of multimedia content [23, 50]. These include metadata for describing still images such as Exif [40], DIG35 [18], and NISO Z39.87 [45]; metadata for describing audio content such as ID3 [43]; and metadata for describing audio-visual content such as P\_Meta [46]. There are also generic multimedia metadata standards such as XMP [66] and MPEG-7 [36], which can be used to describe different types of multimedia content. Moreover, there are general purpose metadata description schemes such as Dublin Core [19].

From these standards, only a few are suitable for describing 3D content. These include the general purpose Dublin Core standard as well as the XMP and MPEG-7 standards. In this section, these metadata standards are presented.

### 2.2.1 Dublin Core

*Dublin Core (DC)* metadata schema consists of a collection of elements designed to foster consensus across disciplines for discovery-oriented description of diverse resources in an electronic environment. The first element set—*Dublin Core Metadata Element Set (DCMES)* [61]—was created during metadata workshop, held in Dublin, Ohio USA [21]. Originally, the DCMES was created to enhance searching of document-like objects on the Web. Nevertheless, it can be applied to any kind of electronic resources, e.g., graphics, sounds, and video files. Due to its generic nature and portability, DCMES quickly became a widely used metadata standard for digital library applications. Currently, work on DCMES specification and extensions is lead by Dublin Core Metadata Initiative (DCMI) [20].

The Dublin Core Metadata Element Set [19] has been formally endorsed by ISO Standard 15836-2003 [39] and ANSI/NISO Standard Z39.85-2007 [44]. DCMES is composed of fifteen elements describing basic properties of content (Table 2.4). The DCMES specification defines only URIs, names, and definitions of the metadata elements. It does not provide information about structure of element values nor their syntax.

Since DCMES is intended to support cross-discipline resource discovery, it does not satisfy all metadata requirements of different application domains. Therefore, in most applications it is used in connection with more advanced, domain-specific metadata standards. DC elements are usually a subset of resource metadata and are used as a minimal metadata for data exchange and discovery.

**Table 2.4** Dublin Core Metadata Element Set reference (version 1.1)

Element	Definition
1. Contributor	An entity responsible for making contributions to the resource.
2. Coverage	The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant.
3. Creator	An entity primarily responsible for making the resource.
4. Date	A point or period of time associated with an event in the lifecycle of the resource.
5. Description	An account of the resource.
6. Format	The file format, physical medium, or dimensions of the resource.
7. Identifier	An unambiguous reference to the resource within a given context.
8. Language	A language of the resource.
9. Publisher	An entity responsible for making the resource available.
10. Relation	A related resource.
11. Rights	Information about rights held in and over the resource.
12. Source	The resource from which the described resource is derived.
13. Subject	The topic of the resource.
14. Title	A name given to the resource.
15. Type	The nature or genre of the resource.

### 2.2.2 XMP

*XMP—Extensible Metadata Platform*—specification [66] has been created by *Adobe Systems Inc.* [3] to improve management, search and retrieval, reuse, and consumption of multimedia assets. The XMP specification consists of three main components: data model, storage model, and schema definitions. The data model is derived from RDF and it has been designed as a subset of the RDF data model. The data model provides support for metadata properties attached to a resource. Properties have property values, which can be simple or complex (structure or array). Properties also may have properties offering information about the property value. The storage model describes serialization of metadata to the XML format. Metadata serialized in such a way can be attached to various types of digital resources and processed across different systems and applications. A schema is a collection of predefined sets of metadata property definitions. There are some predefined schemas in XMP such as Dublin Core schema, basic schema, rights management schema, and media management schema. The collection of schemas is extensible—it is possible to extend existing schemas and to define new ones.

### 2.2.3 MPEG-7

*MPEG-7* [42] is an ISO standard [36] developed by the *Moving Picture Experts Group (MPEG)* [55]. The MPEG-7 standard, formally named “*Multimedia Content*

*Description Interface*,” does not target a particular type of multimedia data, but is intended for describing any kind of multimedia content. MPEG-7 standardizes tools that are used to create metadata. These tools include: *Descriptors*, *Description Schemes*, *Description Definition Language (DDL)*, and a number of *Systems tools*.

Descriptor is the basic element of the MPEG-7 metadata. It is used to represent specific features of content. Descriptor specification defines syntax and semantics of feature representation. Examples of descriptors are: a time code for representing duration of a movie, a color histogram for representing image color, and a character string for representing title.

Description Scheme is a structural component of MPEG-7. It defines structure and semantics of relationships between its components, which may be both Descriptors and Description Schemes. With Description Schemes it is possible to describe multimedia content structure and semantics. For example, it is possible to describe a movie, divided into shots, including semantic textual descriptors at the movie level, and technical descriptors such as color, motion, and audio amplitude at the shot level.

Both description tools presented above are represented by using Description Definition Language (DDL). DDL enables creation of new Description Schemes and Descriptors. It permits extending and modifying existing description tools. MPEG-7 DLL is based on W3C XML Schema [58] recommendation. It uses all XML Schema tools and constructs, and adds a few more, such as array or matrix data types.

Systems tools are not related directly to metadata descriptions. They cover the areas of storage and transfer of descriptions, as well as issues related to intellectual property rights.

## References

1. 3D Industry Forum. <http://www.3dif.org/>. Accessed 17 Aug 2011
2. Adobe Reader X. Adobe Systems Inc. <http://www.adobe.com/products/reader.html> (2011). Accessed 17 Aug 2011
3. Adobe Systems Inc. <http://www.adobe.com/>. Accessed 17 Aug 2011
4. Ames, A.L., Nadeau, D.R., Moreland, J.L.: *The VRML 2.0 Sourcebook*, 2nd edn. Wiley, New York (1997)
5. Arnaud, R., Barnes, M.C.: *COLLADA: Sailing the Gulf of 3D Content Creation*. AK Peters, Wellesley (2006)
6. Arnaud, R., Parisi, A.: *Developing web applications with COLLADA and X3D*. Khronos Group whitepaper. [http://www.khronos.org/collada/presentations/Developing\\_Web\\_Applications\\_with\\_COLLADA\\_and\\_X3D.pdf](http://www.khronos.org/collada/presentations/Developing_Web_Applications_with_COLLADA_and_X3D.pdf) (2007). Accessed 17 Aug 2011
7. Bell, G., Parisi, A., Pesce, M.: *The Virtual Reality Modeling Language, Version 1.0. Specification*. <http://www.web3d.org/x3d/specifications/vrml/VRML1.0/index.html> (1995). Accessed 30 Jul 2011
8. Bormans, J., Hill, K. (eds.): *MPEG-21 overview (Version 5)*. ISO/IEC JTC1/SC29/WG11 N5231. <http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm> (2002). Accessed 17 Aug 2011
9. BS Contact, Version 7.219. Bitmanagement Software GmbH. <http://www.bitmanagement.de/en/products/interactive-3d-clients/bs-contact> (2011). Accessed 17 Aug 2011

10. Bulterman, D. et al.: Synchronized Multimedia Integration Language (SMIL 2.1), W3C recommendation. <http://www.w3.org/TR/SMIL2/> (2005). Accessed 17 Aug 2011
11. Carey, R., Bell, G.: The Annotated VRML 2.0 Reference Manual. Addison-Wesley, Reading (1997)
12. Chiariglione, L.: MPEG: achievements and current work. ISO/IEC JTC1/SC29/WG11—coding of moving pictures and audio. [http://mpeg.chiariglione.org/mpeg\\_general.htm](http://mpeg.chiariglione.org/mpeg_general.htm) (2001). Accessed 17 Aug 2011
13. COLLADA. Approved by the Khronos Group as open standard. Press release. <http://www.scei.co.jp/corporate/release/pdf/050729e.pdf> (2005). Accessed 17 Aug 2011
14. COLLADA—digital asset and FX exchange schema. <http://www.collada.org/>. Accessed 17 Aug 2011
15. Cortona3D Viewer, Version 7.0. Parallel Graphics Ltd. <http://www.cortona3d.com/Products/Viewer/Cortona-3D-Viewer.aspx> (2011). Accessed 17 Aug 2011
16. Dassault Systemes. 3DVIA player. <http://dl.3dvia.com/software/3dvia-player/> (2010). Accessed 30 Jul 2011
17. Dassault Systemes. 3D XML—a universal, lightweight XML-based format. <http://www.3ds.com/products/3dvia/3d-xml/overview> (2011). Accessed 30 Jul 2011
18. Digital Imaging Group: DIG35 specification—metadata for digital images, Version 1.1 (2001)
19. Dublin Core Metadata Element Set, Version 1.1. <http://dublincore.org/documents/dces/> (2010). Accessed 30 Jul 2011
20. Dublin Core Metadata Initiative. <http://dublincore.org/>. Accessed 30 Jul 2011
21. Dublin Core Workshop DC-1: OCLC/NCSA Metadata Workshop: The Essential Elements of Network Object Description. <http://dublincore.org/workshops/dc1/>. Accessed 30 Jul 2011
22. GPAC Multimedia Open Source Project. <http://gpac.wp.institut-telecom.fr/>. Accessed 17 Aug 2011
23. Hausenblas, M. et al.: W3C incubator group report. Multimedia vocabularies on the Semantic Web. <http://www.w3.org/2005/Incubator/mmsem/XGR-vocabularies/> (2007). Accessed 17 Aug 2011
24. Instantreality Framework 2.0. Fraunhofer IGD. <http://instantreality.de/home/> (2011). Accessed 17 Aug 2011
25. International Electrotechnical Commission. <http://www.iec.ch/>. Accessed 30 Jul 2011
26. International Organization for Standardization. <http://www.iso.org/>. Accessed 30 Jul 2011
27. International Standard ISO/IEC 14772-1:1997. International Standard: Information technology—Computer graphics and image processing—The Virtual Reality Modeling Language (VRML)—Part 1: Functional specification and UTF-8 encoding. <http://www.web3d.org/x3d/specifications/> (1997). Accessed 30 Jun 2011
28. International Standard ISO/IEC 14772-1:1997/Amd. 1:2003. International Standard: Information technology—Computer graphics and image processing—The Virtual Reality Modeling Language—Part 1: Functional specification and UTF-8 encoding—Amendment 1—Enhanced interoperability. <http://www.web3d.org/x3d/specifications/vrml/VRML97Am1/> (2003). Accessed 30 Jun 2011
29. International Standard ISO/IEC 14496-1:2010. Information technology—Coding of audiovisual objects—Part 1: Systems (2010)
30. International Standard ISO/IEC 14496-2:2004. Information technology—Coding of audiovisual objects—Part 2: Visual (2004)
31. International Standard ISO/IEC 14496-11:2005. Information technology—Coding of audiovisual objects—Part 11: Scene description and application engine (2005)
32. International Standard ISO/IEC 14496-16:2009. Information technology—Coding of audiovisual objects—Part 16: Animation Framework eXtension (AFX) (2009)
33. International Standard ISO/IEC 14496-20:2008. Information technology—Coding of audiovisual objects—Part 20: Lightweight Application Scene Representation (LAsER) and Simple Aggregation Format (SAF) (2008)
34. International Standard ISO/IEC 14496-21:2006. Information technology—Coding of audiovisual objects—Part 21: MPEG-J Graphics Framework eXtensions (GFX) (2006)

35. International Standard ISO/IEC 14496-25:2011. Information technology—Coding of audio-visual objects—Part 25: 3D Graphics compression model (2011)
36. International Standard ISO/IEC 15938-1:2002. International Standard: Information technology—Multimedia content description interface—Part 1: Systems (2002)
37. International Standard ISO/IEC 19775-1:2008. International Standard: Information technology—Computer graphics and image processing—Extensible 3D (X3D). <http://web3d.org/x3d/specifications/> (2008). Accessed 30 Jun 2011
38. International Standard ISO/IEC TR 21000-1:2004. Information technology—Multimedia framework (MPEG-21)—Part 1: Vision, technologies and strategy (2004)
39. International Standard ISO 15836:2009. Information and documentation—The Dublin Core Metadata Element Set (2009)
40. Japan Electronics and Information Technology Industries Association Standard—JEITA CP-3451B. Exchangeable image file format for digital still cameras, Exif Unified Version 2.3. [http://www.jeita.or.jp/japanese/standard/book/CP-3451B\\_E/#page=1](http://www.jeita.or.jp/japanese/standard/book/CP-3451B_E/#page=1) (2010). Accessed 17 Aug 2011
41. Koenen, R.: Overview of the MPEG-4 standard, V.21. ISO/IEC JTC1/SC29/WG11—Coding of moving pictures and audio. <http://mpeg.chiariglione.org/standards/mpeg-4/mpeg-4.htm> (2002). Accessed 30 Jun 2011
42. Martinez, J.M. (ed.): MPEG-7 overview, Version 10. <http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm> (2004). Accessed 17 Aug 2011
43. Nilsson, M.: ID3 tag, Version 2.4.0. Informal standard. <http://www.id3.org/id3v2.4.0-frames> (2000). Accessed 17 Aug 2011
44. NISO American National Standard. ANSI/NISO Z39.85-2007. The Dublin Core Metadata Element Set. [http://www.niso.org/apps/group\\_public/project/details.php?project\\_id=57](http://www.niso.org/apps/group_public/project/details.php?project_id=57) (2007). Accessed 17 Aug 2011
45. NISO American National Standard. ANSI/NISO Z39.87-2006. Data dictionary—Technical metadata for digital still images. [http://www.niso.org/apps/group\\_public/project/details.php?project\\_id=69](http://www.niso.org/apps/group_public/project/details.php?project_id=69) (2006). Accessed 17 Aug 2011
46. P\_META 2.0 Metadata Library. EBU TECH 3295-v2. <http://tech.ebu.ch/docs/tech/tech3295v2.pdf> (2007). Accessed 17 Aug 2011
47. Pereira, F., Ebrahimi, T. (eds.): The MPEG-4 Book. Prentice Hall, New York (2002). ISBN: 978-0130616210
48. Pincoast Software. Multi-format 3D translator for VRML, X3D, and Collada. <http://www.pincoast.com/swtrans.php> (2011). Accessed 17 Aug 2011
49. Preda, M. (ed.): 3D graphics compression model. ISO/IEC JTC 1/SC 29/WG11 N9892. <http://mpeg.chiariglione.org/technologies/mpeg-4/mp04-gcm/index.htm> (2008). Accessed 17 Aug 2011
50. Smith, J.R., Schirling, P.: Metadata standards roundup. IEEE Multimed. **13**(2), 84–88 (2006)
51. Standard ECMA-363. Universal 3D file format, 4th edn. <http://www.ecma-international.org/publications/standards/Ecma-363.htm> (2007). Accessed 17 Aug 2011
52. Swaminathan, V., Bourges-Sévenier, M.: MPEG-J white paper. ISO/IEC JTC 1/SC 29/WG 11 N7410. <http://mpeg.chiariglione.org/technologies/mpeg-4/mp04-j/index.htm> (2005). Accessed 17 Aug 2011
53. Synchronized Multimedia. World Wide Web Consortium (W3C). <http://www.w3.org/AudioVideo/>. Accessed 17 Aug 2011
54. The Khronos Group: Open standards for media authoring and acceleration. <http://www.khronos.org/>. Accessed 17 Aug 2011
55. The MPEG home page. <http://mpeg.chiariglione.org/>. Accessed 18 Aug 2011
56. The National Museum of Agriculture and Agricultural-Food Industry in Szreniawa. <http://www.muzeum-szreniawa.pl/>. Accessed 30 Jul 2011
57. Understanding MPEG-4: technologies, advantages, and markets. An MPEGIF white paper. The MPEG Industry Forum (2005)
58. W3C recommendation XML schema part 0: primer second edition. <http://www.w3.org/TR/xmlschema-0/> (2004). Accessed 30 Jul 2011

59. Watkinson, J.: *The MPEG Handbook*, 2nd edn. Focal Press, Burlington (2004). ISBN: 978-0240805788
60. Web3D Consortium. <http://www.web3d.org/>. Accessed 30 Jul 2011
61. Weibel, S., Godby, J., Miller, E., Daniel, R.: *OCLC/NCSA Metadata Workshop Report, DC-1: OCLC/NCSA Metadata Workshop*. March 1–3, 1995, Dublin, Ohio, USA. <http://dublincore.org/workshops/dc1/report.shtml> (1995). Accessed 30 Jul 2011
62. Wernecke, J.: *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor, Release 2*. Addison-Wesley, Boston (2004)
63. Wernecke, J.: *The Inventor Toolmaker: Extending Open Inventor, Release 2*. Addison-Wesley Professional, Boston (2004)
64. X3D and related specifications. Web3D Consortium. <http://www.web3d.org/x3d/specifications/> (2011). Accessed 18 Aug 2011
65. Xj3D project. Web3D Consortium. <http://www.xj3d.org/>. Accessed 17 Aug 2011
66. XMP Specification. Adobe Systems Inc. <http://www.adobe.com/devnet/xmp.html> (2010). Accessed 30 Jul 2011