

# X3D: 新一代 WEB 交互式三维图形规范

## X3D: The New Generational Interactive 3D Graphic Specification

林冬梅 (佛山科学技术学院信息中心 528000)

**摘要:**随着网络的发展,虚拟现实造型语言(VRML)的缺点逐渐暴露出来,阻碍了面向 WEB 的交互式三维图形技术的深入发展。因此,出现了下一代面向 WEB 的交互式三维图形规范 X3D,它具有小型化、可扩展、与 WEB 上的多种标准交互操作和媒体紧密集成的特性。本文主要介绍从 VRML 发展到 X3D 的必然性,X3D 的新的结构以及新的设计目标,并详细分析了 X3D 的关键技术:组件化思想及 XML 在 X3D 中的运用。

**关键词:**可扩展的三维图形规范(X3D) 虚拟现实造型语言(VRML) 可扩展标记语言(XML) 组件

### 1 VRML 到 X3D 的发展

VRML (Virtual Reality Modeling Language) 是一种描述三维形体和交互环境的文件格式,所建造虚拟场景是由具有层次结构的多个子节点所描述的各个子场景组成,而生成场景的文件是按照 VRML 规范组织成的文本文件即 VRML 文件。VRML 的访问方式是基于客户/服务器模式,服务器提供 VRML 文件及支持资源,客户通过网络下载希望访问的文件,利用客户端机器高性能的 ASCII 和三维图形处理功能执行代码,对图形渲染,通过本地平台上的 VRML 浏览器交互式访问该文件描述的虚拟境界。VRML 空间比较广阔并且本身具有交互性,VRML 提供了 6+1 度的自由,可以沿着三个方向移动,也可以沿着三个方向旋转,同时还可以建立与其他 D 空间的超连接,所以 VRML 是超空间的。

虽然 VRML 在网上实现三维、可交互的虚拟世界,在很大程度上促进了网络 3D 技术的发展,但随着 WEB 技术的迅速发展,再加上自身的发展缓慢(最后的标准是 VRML97),VRML 的缺点逐渐暴露出来,体现在以下几方面:

(1) 性能表现。首次运行时等待时间过长运行时占用资源过多。画面不连续,图像质量不高,不能满足对图像质量要求很高的应用,软件的一致性差,浏览器之间不兼容。

(2) 编程能力。用 VRML97 提供的节点、字段、事件来直接编程,要表现复杂场景非常困难,而且提供的 API 远不能满足应用程序开发的要求,文件采用 SCII 格式,传输缓慢。

(3) 软件实现。由于 VRML97 过于庞大,规范中定义的一些特性很少使用,而为了满足这些特性,只能不断增加浏览器的大小和复杂性。另外 VRML97 是一个单一的封闭标准,与现有的 Web 技术和多媒体技术的集成能力弱。这些严重阻碍了 Web 交互式三维图形技术的发展,以及交互式三维图形技术在 Web 上的大规模应用,因此就要求

VRML97 不断的发展,从而从 VRML97 发展到 X3D 是必然的,它的转变就是要使 X3D 能更加方便的扩展,尽可能快、尽可能多的吸收更多先进的技术。

X3D (eXtensible 3D-可扩展 3D)是由 WEB3D 联盟于 1998 年底提出的一种新一代三维图形规范。定义了如何在多媒体中整合基于网络传播的交互三维内容。X3D 致力于建立一个三维图形与交互多媒体的统一交换格式,这种格式要能达到像 HTML 或 XML 那样的标准性和可扩展性。它与 VRML97 向后兼容,即能用 X3D 技术浏览 VRML 的内容。

### 2 X3D 的组成及设计目标

X3D 从逻辑结构来讲包括四个部分:内核(核心特性集),VRML97 特征集,应用程序接口和扩展集,如图 1。

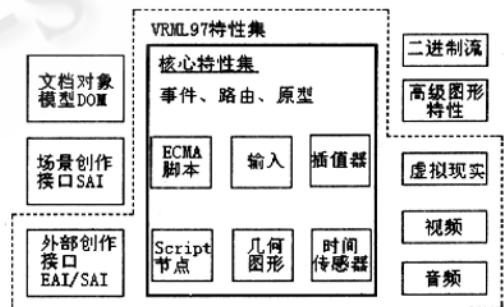


图 1 X3D 的组成

#### 2.1 内核

定义了 VRML 中最为关键的特性,形成核心组件,并将其封装为一个小型的、可扩展的内核,减少实现的复杂性,因而改善了执行过程的可维护性。因此当前的核心特性集只实现 VRML97 的 54 个节点中的 23 个节点。

## 2.2 VRML97 特性集

实现内核以外的 VRML97 元素。VRML97 的节点被设计为可插拔的组件,通过扩展内核,完整地实现了 VRML97 规范定义的功能,从而确保了 X3D 与已有的 VRML 应用兼容。

## 2.3 应用程序接口

由于使用 XML 编码,文档对象模型(DOM)自动为 X3D 提供了一组 API,外部应用程序可以通过 DOM 访问 X3D 文件;作为另一个选择,场景创作接口(SAI)是来自 VRML 的场景图像接口,它将外部创作接口(EAI)和 VRML97 规范中的脚本接口(Script)融合,为外部或内部编程提供了一个单一的程序接口。

## 2.4 扩展集

通过在内核之上进行特性集扩展,实现复杂的或是用户自定义的功能。用户可以在内核之上建立一个完整的 VRML97 特性集。也可以添加其他的扩展,如 NURBS 扩展,二进制文件格式扩展等。通过扩展还可以利用 VRML97 规范中未定义新的硬件渲染技术。组件化的设计为 X3D 的内核提供了一种插件机制,允许扩展功能被实时加入运行内核。

从 X3D 的组成可以看出,其基本的技术特点在于使用当今主流的技术——XML 技术与组件思想——对传统的 VRML 规范进行改造。由于有了 XML 的加入,提高了对事物对象的描述能力,拓宽了 VRML 的使用范围。同时,组件思想在 X3D 的实现中贯穿始终,使得开发手段更加符合现在主流 IT 界的需求。

X3D 为满足特定的市场和技术开发的需要,采用了以下设计目标:

- (1) 分离的数据编码和运行时结构。
- (2) 支持多种数据编码格式,包括 XML。
- (3) 增加新的绘图对象、行为对象和交互对象。
- (4) 给 3D 场景提供可选的应用程序界面(APIs)。
- (5) 尽可能添加完善规格中行为的定义或描述。
- (6) 定义规格的子集“概貌(Profiles)”以满足不同的市场需求。
- (7) 允许在不同层次(Level)的服务上都能实现 X3D 规格。

X3D 被设计为可以在不同的硬件设备中使用,并可用于不同的应用领域中。比如工程设计、科学可视化、多媒体再现、娱乐、教育、网页、多用户共享虚拟世界等方面。

## 3 X3D 的组件化

组件思想来源于面向对象理论,在最新的 X3D 规范中,

使用了组件思想对 X3D 规范进行实现,以增强与其他编程语言的交互。X3D 组件化的设计使得开发者可以容易地加入更多的新特性,也可以在现有的特性上加上新的扩展。可扩展的结构允许 VRML Working Groups 在内核之上浏览器的基础上加入新的规格。因此,X3D 除了将 VRML97 原有的节点包装成组件外,还添加了新的功能组件,其中包括 H-Anim, GeoVRML, Java-DIS-VRML 等。

H-Anim 标准制定了使用人性化动画的方法。这个标准用来建立人性化的矢量动画。H-Anim 可以使用关键帧或 IK 反向动力学等其他技术来执行动画。已经有很多不同的软件以不同的方式实现了 H-Anim 标准,如 Shout3D。GeoVRML 是基于 VRML 的开放源代码的地理信息系统平台规范。为了满足地理信息系统的要求,GeoVRML 添加了特殊的节点以支持多种常用地理坐标系和参考椭球,也可以利用接口与高级语言(Java、C++ )编写的系统进行通信。Java-DIS-VRML 规范同样也是一个在 VRML97 中通过接口使用相应的 Java 程序或脚本的应用。通过 DIS 可以利用现有的协议标准和程序资源在 VRML 中建立高性能、大规模、分布式的多用户实时三维交互系统。

组件是一个个独立的基本功能的模块,例如内核组件、脚本组件、H-Anim 组件等。同时组件内可以设置为不同的层,以支持不同的节点集,或在具体节点内的不同的域的支持。例如 Box 节点在几何组件层 1 中不支持,而在几何组件层 2 中支持。应用程序可以选择最少的组件和最小的层来满足简化程序的需要。同时组件和层还提供了扩展规格的可能。当新的几何节点加入某个组件时,或某个节点加入了新的域的支持,则在这个组件中加入相应的新的层来说明规格的变动,添加功能也可以通过添加新的组件来完成。

概貌是组件的集合,X3D 标准列出了一系列的组件和相应组件的支持层,并且预先制定了一些概貌以满足不同的需要,有了概貌就不需要列出所有的组件。当新的组件和组件新的层被加入 X3D 标准中时,也可以制定某个新的概貌。

概貌、组件和层是一种使 X3D 功能取舍标准化或功能扩展标准化的方式。有了概貌和组件的机制,使用 X3D 标准的公司可以根据自己的需要来决定支持哪些概貌和组件。X3D 的模块结构意味着公司可以从只支持最基本的功能开始,然后逐渐加入他们认为实用的概貌和组件。而各个公司和联盟提出的新功能也可以方便地加入标准中,以快速地扩展 X3D 的功能。概貌、组件、层的概念将在 X3D 文件头列出。这样就保证在运行 X3D 应用前可以先确认是否满足相应的兼容性和一致性。

X3D 的一个发展就是使 VRML97 组件化。组件化可以让应用者根据不同的需要对 X3D 进行功能取舍,如果需要

某些特殊的功能,可以设计扩展的组件。这时大多的功能可以直接使用现有的资源,而不需要重新设计一个完整的应用程序。另外组件化减少了对资源的占用,X3D的组件模型将问题空间分成一个个小部分,与问题相关的部分独立工作于提供相应服务的组件上,同时组件模型避免了再产生像VRML97规范那样的封闭的标准。

#### 4 X3D 与 XML 技术的结合

XML已经成为互联网上通用的世界语,是一种元标记语言,所谓“元标记”就是开发者可以根据自己的需要定义自己的标记,任何满足XML命名规则的名称都可以标记,这就为不同的应用程序打开了大门;其次XML是一种语义结构化语言,它描述了文档的结构和语义。X3D采用XML作为它的编程语言是因为XML有较强的可移植性,对数据归档和移植提供了较长的生命周期。X3D需要能和下一代的网络技术紧密结合,而XML正好是下一代的网络技术的趋势。在开发页面或应用程序时,基于XML整合的系统,场景内容和执行都变得简单,而开发也能更高效。

以下的场景是采用典型的XML编码的X3D文件:

```
<? 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 profile="IMMERSIVE"
xmlns:xsd = http://www.we.org/2001/
XMLSchema-instance
xsd:noNamespaceSchemaLocation="http://
www.web3d.org/specifications/x3d-3.0.xsd">
<head>
<meta name='filename' content='RedSphere-
BlueBox.x3d' />
<meta name='description' content='X3D
scene header syntax example' />
</head>
<Scene>
<Transform>
<NavigationInfo headlighr='false'
avatarSize='0.25 1.6 0.75 type='EXAM-
INE' />
<DirectionalLight />
<Transform translation='3.0 0.0 1.0'>
<Shape>
```

```
<Sphere radius='2.3' />
<Appearance>
<Material diffuseColor='1.0 0.0 0.0' />
</Appearance>
</Shape>
</Transform>
</Transform>
</Scene>
</X3D>
```

上面的XML文档中,在文件的开头先定义了概貌、组件、元数据组,还包括了XML应用中需要使用的DTD、Schema文件的位置。象XML这样的语言具有很强的可移植性。XML已经获得了Netscape Communicator和Internet Explorer等浏览器的广泛支持。XML正好是下一代的网络技术的趋势,这使得X3D能更好地和下一代的网络技术紧密结合。

#### 5 结束语

随着网络技术的进一步发展,宽带、高速及平台性能的提升导致人们对更高质量的网络三维资源的需求更加迫切,从而产生了新一代的三维交互网络图形标准X3D,它以XML与组件思想作为核心技术,解决了VRML97的不足,使得X3D除了在虚拟现实领域发挥自己的本领外,在电子商务、可视化仿真、数据库可视化、娱乐、教育等领域也都有很好的应用前景。

#### 参考文献

- 1 Web3D Virtual Reality Modeling Language, International Standard ISO/IEC 4772:200x, September 20, 2000. <http://www.Web3D.org/TaskGroup/x3d/specification>.
- 2 W3C Recommendation, Extensible Markup Language(XML) 1.0 (Second Edition), October 6, 2000. <http://www.w3.org>
- 3 W3C Working Draft, Document Object(DOM) Level 1 Specification (Second Edition), Version 1.0, September 29, 2000. <http://www.w3.org/TR/REC-DOM-Level-1>
- 4 Justin Couch, XJ3 Architecture, March 15, 2001, Web3D. <http://www.Web3D.org/x3d>
- 5 X3D Requirements and Design Teams, X3D Technical Requirements, Version 1.6, February 9, 1999, Web3D