应急演练角色系统①

郭志星[1,2], 廉东本[2], 苏 谟[2]

[1](中国科学院 研究生院, 北京 100049)
[2](中国科学院 沈阳计算技术研究所, 沈阳 110168)

摘 要: 应急演练系统对于安全生产事故的应急管理具有重要的指导意义, 角色系统是应急演练系统的核心基础, 为演练系统场景提供了可交互的对象. 基于应急演练系统需求和架构, 采用面向对象的设计方法设计了一种通用的角色系统结构, 包括角色类的设计、技能算法类的设计以及数据管理类的设计及其应用. 重点阐述了角色系统的主要功能以及角色系统的设计过程, 解决了角色的复用性, 耦合性. 同时, 可以通过数据配置, 动态绑定角色技能和数据属性, 完善并丰富了角色. 最后, 介绍了角色系统的具体应用和优势所在.

关键词: 设计模式; 角色系统; 解耦合; 工厂模式; 演练系统

Emergency Drill Role System

GUO Zhi-Xing[1,2], LIAN Dong-Ben[2], SU Mo[2]

[1](Graduate School of Chinese Academy of Sciences, Beijing 100049, China)
[2](Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

Abstract: Emergency drill system for accidents in production practice of emergency management has important guiding significance, and role system is the core of the drill system, which provides the scene interactive objects for drill system. This paper, which based on the emergency drill system requirements and architecture, uses object-oriented design method to design a universal role system structure, including the design of role class, the design of skills as well as the design of data of the algorithm of management. This paper expounds the main function of role system and designs process of role system and solves the role of the reusability, coupling. At the same time, it can be dynamically bound role skills and data attributes, improve and enrich the role system through the data configuration. Finally, this paper introduces the specific application of the model and advantage of role system.

Key words: design patterns; role system; decoupling; factory patterns; drill system

1 引言

近年来, 随着产业规模扩大, 国家经济快速发展、国家对人民群众生命财产安全日益重视. 在爆发突发事件或者水体污染时, 能够正确的做出事故分析并制定处理方案, 是科学的抢险救灾工作中至关重要的环节[1]. 传统的实际演练方式有着造价高, 污染大、危险性强、进程不易把握、演练内容单一等缺点[2]. 随着计算机虚拟技术的发展, 由计算机整合模型库、案例库、应急知识库、方法库、图形库等数据库综合模拟出灾害爆发场景, 构建出一个应急演练系统, 这样一个系统不仅可以提高桌面演练的真实度、提高演练的覆盖面, 而且因为成本低廉可以重复使用, 为真实演练做好准备, 提高真实演练的效率和效果.

角色系统是应急演练系统的核心基础, 为演练系统场景提供了可交互的对象. 角色系统是应急演练系统的根本, 角色系统设计的好坏事关演练系统能否正常高效的运行. 传统角色系统的设计主要依赖于选用的具体引擎, 不具备角色系统的复用性和扩展性, 不能适应复杂多变的事故演练情况. 综上所述, 如何提高角色的复用性, 如何动态创建角色实体, 如何动态绑定角色技能是

① 基金项目:国家水体污染控制与治理科技重大专项(2012ZX07505004)
收稿时间:2012-08-16;收到修改稿时间:2012-09-10

角色系统设计的核心. 因此, 本文基于应急演练的应用需求以及角色调度过程的分析, 给出了一种角色系统的设计方案来解决这些问题. 本角色系统主要包括角色模型的调配, 以及角色类的设计, 并根据角色类阐述了角色如何更新角色属性和如何动态增减技能的特性. 最后, 本文介绍了角色系统在应急演练系统中的应用.

2 角色模型

模型是角色系统中角色的表现形式, 鉴于应急演练系统的可交互性, 场景全方位可见性, 本系统采用三维模型去表现系统中的各种角色, 全方位展现现实场景.

根据应急演练系统, 系统中应包括: 参与人员、参与各种车辆、房屋建筑、危险源、水、火、气体、天气情况, 根据这些物体, 具体模型可采用三维模型去仿真: cal3d 骨骼模型、ive 模型、粒子效果. 房屋建筑等一些静态物体可采用 ive 模型去仿真; 人、车辆等一些动态物体可采用 cal3d 骨骼模型去仿真; 水火气体、天气则可采用粒子效果仿真模拟.

三维建模方面, 采用 3dsMAX 建模, 在 3dsMAX 中构建整体演练系统的场景模型, 然后通过 osgExp 模型导出插件输出为 IVE 格式, 这样可以大幅提高场景的加载和渲染速度; IVE 格式是一种 OSG 二进制场景格式, 优点是可以保存几乎所有的 OSG 场景信息(模型, 纹理, 状态, 动画, 着色器代码等), 并且采用二进制格式, 体积小于与之功用相同的.osg 文件.

人物车辆等动态物体也是通过在 3dsMAX 中建好模型, 贴上相应的纹理图片, 通过 Cal3D 导出插件, 输出为程序员可用的 Cal3D 格式文件: 主要包括三类数据: 骨骼数据(扩展名为 csf 或 xsf)、网格数据(扩展名为 cmf 或 xmf)和动作数据(扩展名为 caf 或 xaf), 由一个文本配置文件(扩展名为 rbody)将这些数据文件整合起来,程序加载虚拟人物时是通过加载配置文件来实现的. 模型加载后即可通过简单的函数调用驱动它做出某些预设动作, 在场景中通过程序实现控制.

粒子效果主要通过粒子系统来实现. 通过开源引擎 OSG 中提供了专门的粒子系统工具, 可以简单实现粒子爆炸、火的模拟以及对水、雨雪效果的模拟.

3 角色类设计

根据应急演练系统的实际情况, 角色具备: 角色基本属性、角色扩展属性、角色技能、角色算法. 角色基本属性, 包括所有角色的共性, 如关联模型、位置、大小、包围盒、可选中、本地或者远程等属性; 扩展属性, 是不同角色差异化的表现, 如消防员具有喷水枪, 消防车具有水箱等特性; 角色技能, 是角色具备的功能, 如消防员具有喷水、移动技能, 消防车具有载人、喷泡沫等技能; 角色算法, 是描述角色技能如何改变角色属性的, 即角色属性变化的规则.

本文设计的角色类是角色系统的主要组成部分, 上文提供的角色模型是角色类中一个属性即关联模型. 角色类共包括实体角色类、数据管理类和技能算法类, 实体角色类拥有角色基本属性, 数据管理类拥有角色的扩展属性, 技能属性类拥有角色的技能及技能属性, 三者相互协作, 共同表现演练系统中的角色.

鉴于角色系统应该具有可扩展性、可维护性、可复用性, 角色系统采用设计模式中的工厂模式、代理模式、单例模式.

角色系统体系中, 有几个独一无二的部件, 如果采用常规的实例化对象的方法, 即 New 直接实例化对象, 这样的全局变量会遇到两个问题: 具有较高的耦合性, 缺乏灵活度; 类本身实例化的主动权不被类自身掌握. 为避免这两个问题, 我们采用 Singleton 模式. Singleton 模式保证类仅有一个实例对象, 并提供全局唯一访问点. 但是, 本角色系统体系中有几个 Singleton, 为了避免代码重复, 就有必要另外定义个一个类, 来作为这些 Singleton 的基类. 由于要实现不同类型的 Singleton 的基类却又不能预先知道子类的类型, 我们引入模板技术, 实现模板单例: Singleton<T>.

同样, 角色系统中, 很多对象的创建都是根据某种类型进行创建, 这样的话可以利用工厂模式将其具体的构造函数封装起来, 通过传递类型参数进行创建, 避免了每次都要调用其构造函数. 基于和 Singleton 同样的考虑, 我们可以采用模板技术, 实现模板工厂: ObjectFactory<T>.

代理模式是对象结构模式. 代理模式给某一个对象提供一个代理对象, 并由代理对象控制对原对象的引用. 在一些情况下, 一个客户不能直接引用一个对象, 而代理对象可以在客户端和目标对象之间起到中介的作用. 角色系统体系中, 客户端不直接操作角色, 而是通过角色代理这个中介去操作角色, 由角色代理去控制角色的引用.

根据需求, 需要设计以下几个类: 实体角色类、数

据管理类、技能算法类.

3.1 实体角色类

Actor 是实体角色类的基类, 拥有角色的共性即角色基本属性, 如: 位置、旋转、渲染、包围盒、可否选中、接受演练系统管理及接收演练系统消息等特性; ActorProxy 是角色的代理类, 系统必须通过 ActorProxy 访问 Actor 的基本属性; ActorType 是角色类型, 标明角色的类型, 目前根据角色添加模型的类别, 对角色进行分类, 一种是 IVE 模型, 另一种为 Cal3D 骨骼模型; ActorFactory 是角色工厂, 根据角色类型, 创建角色代理, 通过代理访问角色.

角色代理除了访问角色之外, 并且具有如下通用功能:

1. 给角色添加模型;

2. 设置角色为远程或者本地角色, 判断该角色是否为远程角色.

角色工厂通过角色类型创建角色的时候, 会给每个角色实例产生一个唯一角色 ID 号, ID 号全局唯一, 可以唯一标识此角色实例.

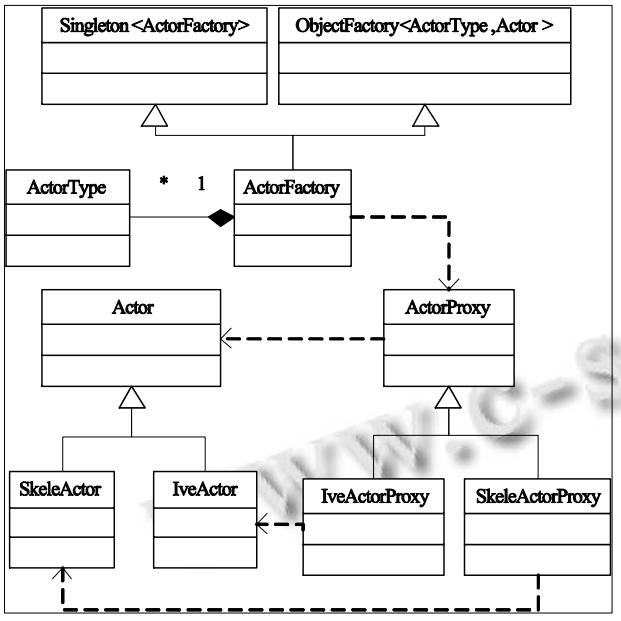

图 1 角色类图

可见, 实体角色类的设计, 可以通过扩展角色类型, 来增加我们的角色的种类; 可以通过维护我们的角色代理来进一步维护我们角色.

3.2 数据管理类

角色类 Actor 只包含了角色的基本属性, 我们还需要一个类来管理 Actor 的其他数据: 扩展属性、技能、算法. 我们可以在数据库中为每个角色实例配置角色的扩展属性和技能算法数据, 在系统初始化的时候, 通过一个数据管理类把角色的各种数据读到内存中. 根据角色 ID 号来查找属于该角色的各种数据. 这样就可以实现角色与数据在代码的分离.

根据上边所描述, 系统初始化数据类所需数据结构: map< 角色 ID, vector< 扩展属性>>, map< 角色 ID, vector<技能 ID>>, map<map<角色 ID, 技能 ID>, Vector< 算法 ID>>, 存储每个角色的扩展属性和技能,和技能对应的算法. 由于本类主要是存储数据, 可作为单例, 全局唯一, 并提供获取这些数据的接口, 设计如图 2 所示.

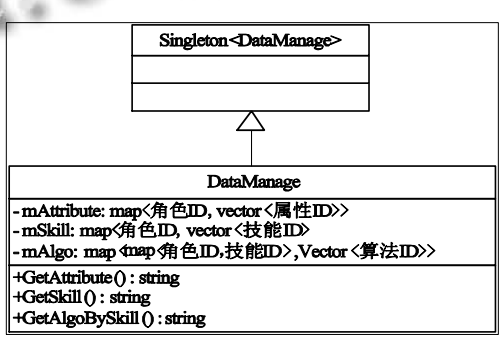

图 2 数据管理类图

可见, 数据管理类的使用, 我们可以比较容易的在外部为角色添加扩展属性, 并且较为容易的对角色数据的统一管理.

3.3 技能算法类

Skill 是角色技能的基类, 涵盖角色各种技能, 如: 爆炸、喷水、移动、治疗等技能; Algo 为角色算法的基类, 主要用来实现技能的功能; AlgoType 是算法类型; AlgoFactory 是算法工厂, 根据算法类型, 创建算法, 设计类图如图 3 所示.

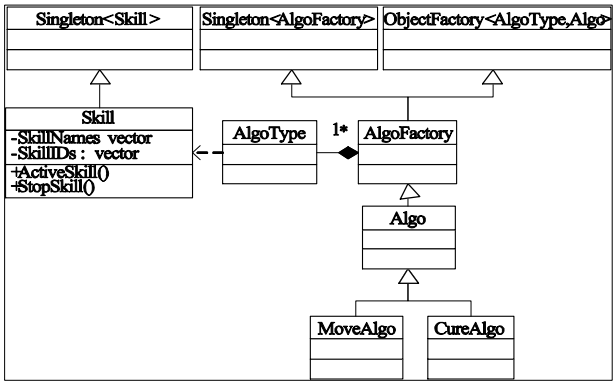

图 3 技能算法类图

算法依赖于技能，一般来说，每个技能可能对应多个算法，比如消防员灭火技能对应水量的变化算法、消防员体力的变化算法、被灭物体生命值变化的算法.

当角色释放技能时，可以确定处理该技能的算法类型，算法工厂根据算法类型创建相应的算法实现该技能的释放效果以及通过该算法改变角色的相应属性.

4　角色系统应用

角色类的设计不仅使角色的基本属性和扩展属性得到了很好的分离，在数据库中可以通过配置增加角色的扩展属性，而且角色的技能具有了通用性，可以把技能作用于任何一个具体角色上，使角色系统得到了很好的扩展.

当用户启动演练系统时，数据管理类首先要从数据库中读取角色实例的属性和技能算法数据，完成数据管理类的初始化. 用户选中某个角色，释放某个技能，首先通过消息激活角色的某个算法，算法经过数值计算，通过消息去驱动角色的基本属性和扩展属性的改变. 如此反复在系统中运行，直到用户关闭演练系统，流程如图 4 所示.

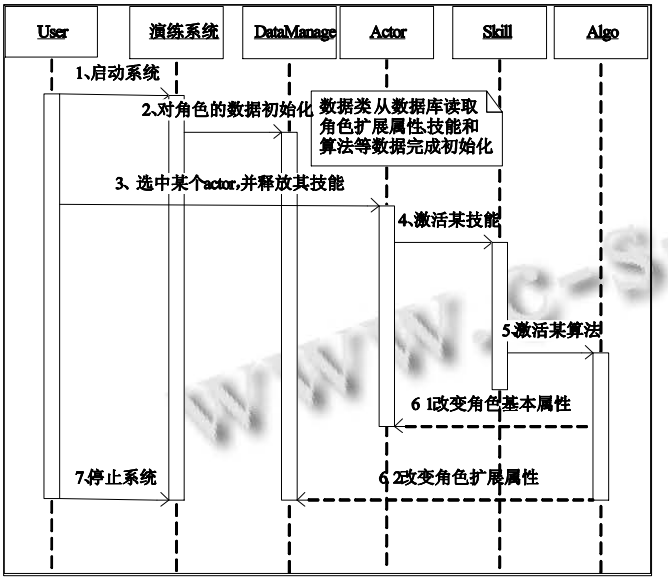

图 4　角色运行时序图

角色系统已成功运用于应急演练系统，图 5 是在数据库中配置角色数据. 图 6 是根据配置在场景中的应用. 由此可见，角色系统很好的应用于应急演练系统中.

图 5　数据配置

图 6　角色应用

5　结语

诸多设计模式的使用，使得角色系统的可维护性和可复用性得到了很大提高. 角色的设计不再依赖于具体引擎，可以适应复杂多变的演练场景，可以根据具体场景给配置角色的模型和角色的技能和算法，而这些不必修改代码，只要在数据库中进行正确配置即可.

参考文献

1 孙成江,刘林.应急救援模拟演练系统设计与实现初探.石油工业计算机应用,2010,(3):3–6.

2 熊珺.国民经济动员仿真演练系统的任务分配研究.华中科技大学,2006.

3 孙咏.基于 OCP 软件应用架构的设计与实现[博士学位论文].北京:中国科学院研究生院,2009.

4 阎宏.JAVA 与模式.北京:电子工业出版社,2002.41–44.

5 Erich Gamma 设计模式-可复用面向对象软件基础(中译本). 北京:机械工业出版社,2000.1–21.

6 Shalloway A. Design patterns explained. Pearson Education, 2002.

7 Berenbrink P, Brinkman A, Scheduler C. SIMLAB—A Simulation Environment for Storage Area Networks. IEEE, 2001,(4):227–234.