

动画剧本描述语言 SDL/A 的设计与实现*

马华东 刘慎权

(中国科学院计算技术研究所 CAD 开放实验室 北京 100080)

唐小平 赵琛

(中国科学院软件研究所 北京 100080)



摘要 本文介绍了基于时序逻辑的动画描述模型和基于此模型设计的动画剧本描述语言 SDL/A。这种语言具有便于动画设计各个层次描述、能够描述设计的逐步求精过程、能描述动画中的各种抽象对象以及角色间动作的同步等特点,并易于将这种基本的通用的剧本描述语言集成到一个功能强大的 CASE 环境—XYZ 系统之中。本文主要介绍了 SDL/A 语言的设计思想和实现技术。

关键词 动画描述模型,时序逻辑,SDL/A,动画语言,同步,逐步求精。

对计算机动画的研究大致可划分为 3 个部分:造型、运动描述与合成及绘制技术。其中动画描述和运动的控制是动画研究的主要问题,目前通用的运动控制方法往往还需要大量的人工干预来指定运动,目前的动画描述方法难以描述诸如多角色动作同步、消息传递、自动路径规划及设计过程等复杂的动画现象。

动画描述模型是控制动画生成的方法,不同的模型反映了不同的系统自主(Autonomy)能力。Zeltzer^[1]定义了三级系统自主能力:在引导(Guiding)级系统中动画师应当预先知道运动对象不同参数的值并指定它们;在程序设计级系统中计算机应存储充分的知识来解释由动画师约定在剧本中的基本命令;在任务级系统动画师仅需给出诸如初始位置和受力情况这样的信息,计算机就能从几何数据库中取得必要的信息并根据物理和机械定律来生成运动。从动画发展前景来看,最高目标应该是动画由自然语言描述的剧本而生成。

人们解决一个大的问题时,往往要经历下述过程:首先由问题的要求到形式描述的过程,称之为功能描述过程;其次是从形式化描述到有效的实现,称之为程序构造。在制作一个动画电影过程中,导演首先从剧本(通常以自然语言编写)开始,构思整个影片的拍摄过程,一般将整个过程划分为一些场景,每个场景由很多镜头构成。在具体镜头中,每个角色的动作将剧本的故事情节表演出来。因此,在用计算机动画系统制作动画片时,对整个动画的描

* 本文研究得到国家自然科学基金和北京邮电大学程控交换技术通信网国家重点实验室资助。作者沈武威,1967年生,助研,主要研究领域为语言转换与面向专用领域的软件工具与环境。唐雅松,1925年生,研究员,中国科学院院士,博士生导师,主要研究领域为计算机科学与软件工程。

本文通讯联系人:沈武威,北京 100080,中国科学院软件研究所

本文 1995-03-22 收到修改稿

述也应包括这个逐步求精的设计过程。

在计算机动画中,一个运动对象定义为其属性随着时间变化的物体,这里的属性包括 2 部分:表示位置、形状、大小等的几何属性;表示颜色、材质、纹理等的物理属性.运动对象在动画中又称之为角色(Actor),它一般包括一般的运动对象(具有形状的物体)和特殊的运动对象(光源和摄像机).在一个复杂的场景中往往包含多个运动对象,因而运动对象间的“并行性”即行为的并发描述是系统应考虑的重要因素。

基于上述分析,对动画设计逐步求精过程,动画中各个抽象对象以及运动对象间的并发描述是动画描述模型应考虑的重要因素,一个理想的动画描述模型应能描述抽象、并发控制以及角色根据环境的决策能力.本文介绍的基于时序逻辑的动画描述模型是探索理想的动画描述模型的一种尝试.根据基于时序逻辑描述模型开发的剧本描述语言 SDL/A (script description language/advanced)旨在提供一个基本的语言内核,为开发一个动画制作环境提供基础。

1 SDL/A 的设计思想

1.1 面向对象技术对动画描述的影响

数据抽象的工作导致抽象数据类型的概念,这个概念对计算机图形学产生了重要的影响.1969年,Kay在他的博士论文中认为一个程序设计系统应当是基于能够互相传递消息的对象,面向对象程序设计最重要的概念有:每个实体是一个对象;每个对象是类中一个实例;对象由一些私有存储和一些操作组成;对象间通过发送消息相互通讯。

在动画控制中最大的问题是产生一个动画序列所必要的大量数据.解决此类问题目前广泛使用的技术是运用抽象,这正是面向对象技术的主要特点.在计算机动画中有用的 5 种抽象是:结构抽象、运动抽象、功能抽象、角色抽象和世界造型.^[2]基于面向对象思想开发了不少动画系统,如 Chua 等人开发的 SOLAR(1988),Breen 等人开发的 CLOCKWORKS (1987),Turner 等人开发的 FIFTH DIMENSION(1990)等。

1.2 Hewitt 的角色理论

Hewitt 将角色(Actor)定义为一个可发送或可接收消息的对象,一个系统中所有元素是角色,系统中仅有可能的活动是角色间的消息传送,程序设计纯粹是告诉不同的角色如何对它们接到的消息做出反应。

角色理论具有以下优点:改变一个程序意味着简单地改变角色的反应;程序复杂性不随着程序增长而增加;它提供了面向应用所需知识的一种自然表示,智能因素作为具有限制个体的集成体进行建模。

Hewitt 的角色就象是一个人可以接受询问消息,发送消息给其它角色和记忆事情.这种思想为表示计算机动画中动态对象提供了一种理想的方法.受角色理论影响而开发的动画语言和系统有:DIRECTOR,ASAS 等。

1.3 基于时序逻辑的动画描述模型

Manna 和 Pnueli 提出了线性时序逻辑系统^[3],它是一阶谓词逻辑的扩充.基于这种线性时序逻辑理论,唐稚松^[4]设计了一个基于时序逻辑且适应多种程序设计方式的 CASE 环

境-XYZ 系统,它的核心是时序逻辑语言 XYZ/E,其目标是提高软件可靠性和软件生产率。^[5,6]时序逻辑是一种很强的描述工具,它适应于:逐步求精的设计描述;抽象描述及其验证;并发进程与数据模块相结合意义下的面向对象的程序设计;逻辑程序设计等多种程序设计风范。基于时序逻辑的这些特点,我们觉得它很适合动画设计过程中的逐步求精的描述,动画对象的动作同步的描述以及各种抽象数据的描述,因而提出一种基于时序逻辑的动画描述模型。^[7]这种模型实际上是一个基于时序逻辑算子而扩充的场景状态转换网络,即在描述动画角色场景状态变换时,加入了时序算子的运算。每个角色在动画中的运动在剧本中通过一个独立的描述单元来描述,这个描述单元的形式为:

```
{ parameter definition;
  [ce1;ce2;...;cen]
} With pa;
```

其中 ce_i 称之为条件元(Condition Element),它的形式是:

$$lb = S_i \wedge P_i = > @_i (Q_i \wedge ACT = act_i \wedge BG = b_i \wedge TR = t_i \wedge lb = S_j)$$

$@_i$ 是时序逻辑算子,包括 \bigcirc ($\$ \bigcirc$, Nexttime), \diamond ($\langle \rangle$, Eventually), \square (Always), \triangleright ($\$ U$, Until). 设有状态序列 $\sigma; S_0, S_1, \dots, S_j, \dots$, 时序逻辑算子定义如下:

Always(\square): $(\sigma, j) \models \square p$, 如果 $\forall k \geq j, (\sigma, k) \models p$;

Eventually(\diamond): $(\sigma, j) \models \diamond p$, 如果 $\exists k \geq j, (\sigma, k) \models p$;

Next(\bigcirc): $(\sigma, j) \models \bigcirc p$, 如果 $(\sigma, j+1) \models p$;

Until(\triangleright): $(\sigma, j) \models \triangleright q$, 如果 $\exists k \geq j, (\sigma, k) \models q$ 并且 $\forall i, j \leq i < k, (\sigma, i) \models p$.

P_i 和 Q_i 都是合式的时序逻辑公式, Q_i 和 act_i 代表当条件 P_i 满足时角色的行为; BG 是场景的背景描述; TR 给出了行为的执行时间; lb 是一个指明当前状态和下一状态的控制变元; With 后的 pa 是子角色的父角色的名字。

2 剧本描述语言 SDL/A

剧本描述语言是用户和动画系统的核心。根据基于时序逻辑的动画描述模型,参考国际上一些有影响的动画语言和流行的商业动画软件^[8,9]以及 XYZ 的核心语言 XYZ/E^[6],我们设计一种新的剧本描述语言 SDL/A,这种语言是由一组描述单元组成,它们分别用来描述动画中的角色、摄像机、背景、光源、变换、坐标系、角色行为、光源行为、摄像机行为等对象,下面我们分别介绍它们的描述方法。

2.1 层次坐标系

为了便于各个对象的描述,每个对象都可在局部坐标系中进行定义。SDL/A 提供大多数图形系统所支持的层次坐标系,它的描述方式如下:

```
Reference {
  World: (Tman) refman, (Tcar) refcar;
  refman: (To) refo;
}
```

这里包含 4 个坐标系: World(世界坐标系), refman, refcar, refo. 其中 refman 由父坐标

系 World 经变换 Tman 得到,refcar 由父坐标系 World 经变换 Tcar 得到,refo 由父坐标系 refman 经变换 To 得到.

2.2 变换

从一个坐标系到另一个坐标系的变换在 SDL/A 中以坐标系的平移参数;相对 X 轴、Y 轴、Z 轴旋转的角度(逆时针为正);在 X,Y,Z 方向坐标的比例变换系数来定义.

2.3 背景

背景描述包括所有属性不变的物体(道具)、固定光源、背景颜色、环境光的设置等内容.其中关键字 Decor 后给出了所有道具的名字和数据文件名字;关键字 Lights 后面给出了所有固定光源的位置、亮度、参考坐标系等参数;关键字 Ambient 和 Backcolor 后面分别给出了环境光和背景颜色的定义.在 SDL/A 中,参考坐标系缺省为世界坐标系.

2.4 角色行为

角色行为描述单元给出动画中角色动作的详细描述.对于角色行为,我们抽象出以下几个描述项:平移参数,旋转参数,变比参数,参考坐标系等.它的描述单元形式为:

```
Actoraactions {
    aaction1;...;aactionn;
}
```

它有若干动作函数构成,每个函数可给出若干虚参变元,对动作参数的描述共有 4 种运动类型 F,L,S,D,分别表示不变、线性变化、样条曲线变化和以数据文件给定变化过程.一个函数描述简单实例如下:

```
amove(float x1,float x2);
T L (x1,300,0->x2,300,0),
Reference F refx;
```

2.5 摄像机行为

摄像机的行为描述单元给出动画中摄像机动作的详细描述.对于摄像机行为,我们抽象出以下几个描述项:位置、目标点、镜头范围、参考坐标系等.它的描述单元形式与角色行为描述单元基本一致;每个动作函数的描述形式和角色的动作函数描述方法类似.

2.6 光源行为

光源的行为描述单元给出动画中光源运动的详细描述.对于光源行为,我们抽象出以下几个描述项:位置参数、亮度参数、参考坐标系等.它的描述单元形式也与角色行为描述单元基本一致;每个动作函数的描述形式和角色的动作函数描述方法类似.

2.7 角色

角色描述是动画描述的主体,其描述单元包括参数描述部分和规则描述部分.参数描述包括通道定义、运动对象的数据文件以及参考坐标系定义等;规则描述部分以条件元形式定义了角色的运动过程,为了便于描述,此中还可定义局部变量.一个角色描述单元的形式可参见例 1.

2.8 摄像机

在动画中,摄像机也可看作一个特殊的角色加以定义.其描述单元的形式除了个别参数定义外,和角色定义基本一致.

2.9 光源

在动画中,一个属性变化的光源可看作一类特殊的角色加以定义.为了简化描述,我们仅考虑点光源的描述.其描述单元的形式除了个别参数定义外,和角色定义基本一致.

2.10 动画过程控制

动画过程控制单元定义了动画生成时所需的控制参数,包括时间(Tick)区间、速度定义(每个时间单位的帧数)、输出动画帧序列的数据文件等参数.一个描述实例如下:

```
Animation {  
    Tick:0..10;  
    Speed:5;  
    Outfile:ex1.scn;  
}
```

关于 SDL/A 语法的其它描述,限于篇幅不在此一一讨论.

3 SDL/A 的描述特点和实现技术

为了实现 SDL/A 语言,我们开发了一个语言解释器.因为 SDL/A 是一个基本的动画描述语言,它的后继版本将在此基础上进行扩充,以适应更复杂的动画描述和 CASE 制作环境的要求.下面介绍 SDL/A 实现中涉及的几种关键技术.

3.1 通信机制

在 SDL/A 中,每个角色(包括光源和摄像机)的运动行为被独立描述,角色的动作协调是通过消息传递机制来支持.角色间的消息传递是通过 2 个角色之间的通道实现,在 SDL/A 中,通道是可以动态决定的.有 2 条与通道操作相关的通信命令,即输出(写通道)命令 `ch!y` 和输入(读通道)命令 `ch?x`,这里 `ch` 是从输出进程到输入进程之间的通道的名字,`y` 是输出进程的输出表达式,`x` 是输入进程的输入变量.在 SDL/A 中,为了便于实现,我们只考虑同步通信,异步通信方式将在以后的版本中加以考虑.

3.2 子角色

在描述复杂的动画角色时,一个角色描述单元往往难以胜任,如对机器人的描述.在 SDL/A 中,我们引入了一种层次描述角色的方法,并以子角色技术来实现.子角色本身是一个角色,并具有自己独立的描述单元,同时子角色又伴随着它的父角色一起运动,子角色和父角色的动作协调通过消息传递来实现.这种父子角色关系的描述可以发展到任意层次,以描述复杂角色的运动.

3.3 抽象描述

面向对象技术定义各种事情为不同的类和对象,它具有诸如信息隐藏、动态封装、多态性和继承等特点,这种技术较容易地将真实世界模型映射为计算机模型.面向对象的技术应用到图形学中的一个重要领域就是用于动画系统^[6]的开发.在 SDL/A 的开发过程中,我们使用了面向对象的设计思想,为了便于描述和编程,我们将变换、行为、背景、角色、光源、摄像机、规则等定义为动画系统中的类,并且定义这些类的属性和操作.在 SDL/A 中包含着一个类库,它提供了上述类的定义并且是可扩充的.

3.4 逐步求精的描述

时序逻辑的一个主要特点就是支持逐步求精设计的描述和验证,基于时序逻辑的动画剧本描述语言自然应支持动画设计的逐步求精过程.在SDL/A的角色描述体中,提供了4种时序算子的运算,即: $\$O$, $\$U$, $\langle \rangle$, $[\]$,其中后面的2个主要用于设计过程中的抽象性描述.

3.5 语言解释器

剧本解释器的工作过程分为下面几个步骤:

第1步:将SDL/A剧本行为定义部分和角色描述部分通过扫描分析先分别转换成C++描述和并发C描述,这是剧本的预处理过程.这个过程中,首先使用词法分析程序对剧本进行词法分析,验证剧本词法是否有误,借助UNIX系统提供的LEX,我们开发了SDL/A的词法分析程序;词法正确以后,接下来对语法进行分析,通过对每个描述单元每个描述项的分析,得到动作描述的具体含义和每个角色(包括摄像机和光源)的运动过程,然后分别加以转换,这里我们使用YACC开发了一个语法分析程序.

第2步:调用并发C编译器进行剧本中各个角色的并发行为模拟,从而产生一个场景状态转换规则组成的动作原语序列,这个过程称之为中间代码生成过程.所有的信息传递、角色的动作规划和角色间的动作协调都在这个阶段进行解决.这个过程调用了并发C编译器.

第3步:合成所有角色的运动从而求出动画的所有关键状态(帧);根据关键帧间的时间分辨率以及角色的运动类型,可插值生成所有关键帧间的其余中间画面,最后得到一个动画帧数据序列,这个过程称之为动画生成过程.这个过程调用了C++编译器.如果系统图形处理速度很快或者真实感效果要求不高,在每帧数据生成的同时可直接在终端上生成其画面图象,达到实时效果;一般情况下,需要记录这个数据序列并提交动画绘制器逐帧生成动画图象并记录输出.

3.6 绘制技术

动画绘制是生成动画的一个重要步骤.通过实现光线跟踪算法开发了一个动画绘制器.为了加速绘制速度,我们开发一种新的图象绘制技术——跨距扫描线跟踪技术来改进绘制系统,显著提高了绘制速度.

3.7 实现环境

系统实现的硬件平台是SGI系列图形工作站.实现语言和系统用到的软件工具包括:UNIX操作系统;词法分析工具LEX和语法分析工具YACC;面向对象的程序设计语言C++和并发程序设计语言并发C;GL图形库;SGI提供的其它图形图象处理工具.

4 例子

例1:乘出租车问题.大致的情节是:一(机器)人在路上行走,到路口发现一辆出租车,就招呼出租车开过来,乘车离开.

这个过程的关键有2点:①人身体与腿动作的协调;②人与出租车消息传递与同步.用SDL/A语言描述各个角色和摄像机的运动如下.

对人(身体)的描述:

```

Actor man {
  Chan;ch1(car,*),ch2(*,car),ch3(*,lleg),ch4(*,rleg);INT;
  Shape;man.dat;Reference;refTman;
  Rules:[loc[i;int];
    lb=START=>$Oi=0`$Olb=sm1;
    lb=sm1=>$Och3! 8`$Och4! 8`$OAct=walk(0,0,0,800,0,0)
      $OBG=B0`$OTR=8`$Olb=sm2;
    lb=sm2=>$Och2! 1`$Olb=sm21;
    lb=sm21=>$OAct=silence()`$Och1? i`$Olb=sm22;
    lb=sm22=>$Och3! 2`$Och4! 2`$OAct=entercar()`$OBG=B0
      ` $OTR=2`$Olb=sm3;
    lb=sm3=>$Och2! 2`$Olb=sm4;
    lb=sm4=>$Olb=STOP ]
};

```

对两腿的描述,可把两腿看作随人(身体)运动并具有自身运动的子角色,由于左腿和右腿的描述相似,仅给出左腿的描述:

```

Actor lleg {
  Chan;ch3(man,*);INT;
  Shape;lleg.dat;Reference;refTo;
  Rules:[loc[i,j,k;int];
    lb=START=>$Oi=0`$Oj=0`$Olb=sm0;
    lb=sm0=>$Och3? k`$Olb=sm1;
    lb=sm1=>$Oj=j+2`$Olb=sm11;
    lb=sm11=>$OAct=step1(0,0,3.14/4)`$OBG=B0`$OTR=1`$Olb=sm12;
    lb=sm12=>$OAct=step2(0,0,3.14/4)`$OBG=B0`$OTR=1`$Olb=sm13;
    lb=sm13` (j<k)=>$Olb=sm1;
    lb=sm13` (j>=k)=>$Olb=sm2;
    lb=sm2=>$OAct=silence()`$Och3? k`$Oj=0`$Olb=sm3;
    lb=sm3=>$Oj=j+2`$Olb=sm31;
    lb=sm31=>$OAct=step1(0,0,3.14/4)`$OBG=B0`$OTR=1`$Olb=sm32;
    lb=sm32=>$OAct=step2(0,0,3.14/4)`$OBG=B0`$OTR=1`$Olb=sm33;
    lb=sm33` (j<k)=>$Olb=sm4;
    lb=sm33` (j>=k)=>$Olb=sm3;
    lb=sm4=>$Olb=STOP ]
} With man;

```

出租车的描述如下:

```

Actor car {
  Chan;ch2(man,*),ch1(*,man);INT;
  Shape;car.dat;
  Reference;refcar;
  Rules:[Loc[i,j;INT];
    lb=START=>$Oi=0`$Oj=0`$Olb=sc1;
    lb=sc1=>$OAct=silence()`$OBG=B0`$Och2? i`$Olb=sc10;
    lb=sc10=>$OAct=run(0,0,700)`$OBG=B0`$OTR=5`$Olb=sc11;
    lb=sc11=>$OAct=stopcar()`$OBG=B0`$OTR=2`$Olb=sc2;
    lb=sc2=>$Och1! 1`$Olb=sc3;
    lb=sc3=>$Och2? j`$Olb=sc31;
    lb=sc31=>$OAct=run(0,0,300)`$OBG=B0`$OTR=4`$Olb=sc4;
    lb=sc4=>$Olb=STOP ]
};

```

摄影机的描述如下:

```

Camera CAM001 {
  Rules,[
    lb=START=>$Olb=sal;

```

```

lb=sa1=>$○Act=targetisman() $○BG=B0 $○TR=8 $○lb=sa2;
lb=sa2=>$○Act=fixcamera() $○BG=B0 $○TR=16 $○lb=sa3;
lb=sa3=>$○lb=STOP ]
};

```

限于篇幅,上面的描述并不完全,省略了变换、背景、行为等的描述.剧本解释后生成的结果可见图 1.

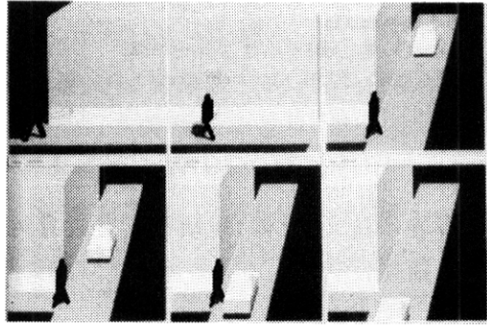


图 1 乘出租车问题

例 2: 台球撞击问题.大致情节是:2 个台球撞击后,先后弹入洞中.其动画过程可见图 2,详细过程不再赘述.

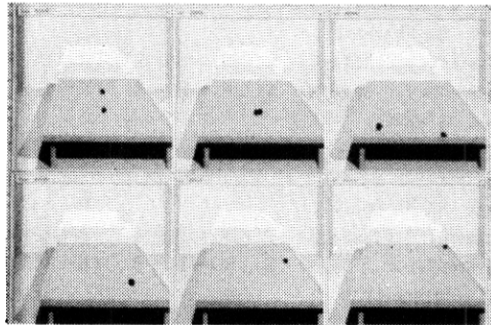


图 2 台球撞击问题

5 结束语

本文首先介绍了动画描述语言 SDL/A 的设计思想和 SDL/A 描述方法,还论述了动画剧本语言解释器的设计和实现策略.基于上述思想,我们实现了剧本描述语言 SDL/A 和一个动画制作环境.进一步工作将从以下几个方面考虑:从描述方面,SDL/A 仅给出了最基本的动画描述,要想使 SDL/A 实用化,还必须考虑在描述功能的全面性上进一步加强,如关于动画设计的逐步求精描述及其验证,描述过程的交互性等;从真实感图象绘制技术方面,目前的动画绘制还没有考虑象纹理等复杂的真实感效果,需在今后的改进中加入这些效果;从设计环境方面,SDL/A 的目标是纳入到 CASE 环境 XYZ 系统之中,并开发一些动画制作需要的工具软件;从应用方面,我们将在完善系统的基础上,力争将 SDL/A 及其制作环境应用到实际动画制作中;另外,我们将尝试把本文的方法应用与动画相关的其它领域中.

致谢 在项目的研究和本文的写作过程中,始终得到了中国科学院院士、中国科学院软件研究所研究员唐稚松先生的指导和帮助,本文的许多思想得益于唐先生的启发,在此向唐先生

表示真诚的感谢.

参考文献

- 1 Zeltzer D. Towards an integrated view of 3D computer animation. *The Visual Computer*, Springer-Verlag, 1985, (1):249~259.
- 2 Hegron G *et al.* Toward general animation control. *Proceedings of CGI'88*, Springer-Verlag, 1988. 54~63.
- 3 Manna Z, Pnueli A. Temporal logic for reactive system. Springer-Verlag, 1991.
- 4 唐稚松. XYZ 的设计思想. *软件学报*, 1990, 1(1):47~55.
- 5 缪旭, 唐稚松. 基于设计的逐步求精方法及环境. *软件学报*, 1990, 1(3):15~27.
- 6 赵琛, 唐稚松. 一种面向软件工程的时序逻辑语言. *软件学报*, 1994, 5(12):1~15.
- 7 Ma Huadong, Liu Shenquan. A simulation oriented animation description model. *Proc. IEEE ICIT'94*, Guang-zhou, Dec. 1994.
- 8 Thalmann N M, Thalmann D. *Computer animation: theory and practice*. Springer-Verlag, 1990.
- 9 TDI. EXPLORE USER'S GUIDE, Volume2.

THE DESIGN AND IMPLEMENTATION OF ANIMATION SCRIPT DESCRIPTION LANGUAGE SDL/A

Ma Huadong Liu Shenquan

(CAD Laboratory Institute of Computing Technology The Chinese Academy of Sciences Beijing 100080)

Tang Xiaoping Zhao Chen

(Institute of Software The Chinese Academy of Sciences Beijing 100080)

Abstract This paper introduces TLAD (temporal logic based animation description) model and an animation description language SDL/A based on this model. SDL/A has powerful description functions, and is convenient to express hierarchy abstract specification and the step-wise refinement procedure of designing an animation. This language can be used to describe multiactor action concurrency. It is easy to be integrated into a powerful CASE environment—XYZ system. This paper mainly introduces the design philosophy and the implementation techniques of SDL/A.

Key words Animation description model, temporal logic, SDL/A, animation language, concurrency, step-wise refinement.