TIPOS DE DATOS ABSTRACTOS PARA LA ESPECIFICACION DE LA ANIMACION: SCRIPTS, ACTORS Y MOVIES

José Alberto Fernández*, Andreas Meier**

RESUMEN

En el presente trabajo se muestran algunos resultados de la aplicación de las técnica de especificación formal, mediante tipos de datos abstractos, al problema de la animación dentro de un sistema de producción de películas sintetizadas por computadora. En este sentido, se llega a la definición de tres tipos principales en el proceso de animación: Los Actores u objetos a ser animados, El Guión (Script) que define la animación y La Película (Movie) resultante.

ABSTRACT

In this paper we present some results concerning the application of the technique of formal specification using abstract data types to the problem of animation in the production of computer synthesized films. To this end, we define three principal types in the animation process: The Actors, or objects to be animated, The Script which defines the animation, and the resulting Movie.

*Profesor de la Universidad Simón Bolívar, Caracas, Venezuela.
Ingeniero de la Computación, USB. Areas de Interes: Computación Gráfica, Lenguajes de Programación e Inteligencia Artificial.

**Profesor de la Universidad Simón Bolívar, Caracas, Venezuela.
Phd en Ciencias de la Computación, Universidad de Ginebra, Suiza.
Areas de Interes: Computación Gráfica, Tratamiento de Imágenes e Inteligencia Artificial.

Uno de los aspectos menos investigados en el area de la Computación Gráfica, es el uso de las técnicas de especificación como base para el desarrollo de software. En realidad, el desarrollo de muchos sistemas se realiza en forma empírica desde el punto de vista de su concepción, con lo cual se llega a productos monolíticos en los cuales es dificil realizar cambios de diseño.

Por otro lado, la descripción imformal que se tiene de estos sistemas, trae como consecuencia, una mayor dificultad en el empleo de los mismos, pues no se posee una fuente de referencia para el implementador o usuario, clara y poco dependiente de una implementación particular.

Este tipo de problemas, ha sido enfrentado con exito en otras áreas de la computación (primordialmente en el área de Lenguajes de Programación) mediante el uso de las diversas metodologías de especificación formal que se han desarrollado.

En Computación Gráfica, se han comenzado a realizar esfuesos en esta vía y se ha llegado a algunos resultados importantes como los obtenidos por Mallgren [Mall82] que utiliza los Tipos de Datos Abstractos y los Tipos de Datos Compartidos (Shared Data Types) para especificar, no sólo los objetos que se manipulan sino tambien las primitivas de interacción que se utilizan en la comunicación usuario-sistema gráfico.

El objetivo de nuestra investigación, se centra en el uso de éste formalismo, en la concepción e implementación del módulo de descripción de la animación del Sistema de Producción de Peliculas Sintetizadas por Computadora que en estos momentos esta siendo desarrollado por el grupo de Computación Gráfica de la Universidad Simón Bolívar.

EL SISTEMA DE ESPECIFICACION FORMAL

El lenguaje formal utilizado, emplea las tecnicas de especificación algebraica de Guttag [Gutt75], Zilles [Zill74] y Goguen el al. [Gogu78] para describir los tipos de datos abstractos y sigue la sintáxis propuesta por Mallgren con ciertas modificaciones que facilitan la definición y uso de tipos de datos parametrizados.

Los operadores básicos de generación se reconocen en la especificación, por estar precedidos por un '·'. Las operaciones auxiliares, por estar precedidas en la firma por un '*'.

TIPOS BASICOS PARA LA DEFINICION DE OBJETOS JERARQUICOS

Debemos definir en primer lugar algunos tipos pictóricos básicos que son necesarios en nuestro sistema gráfico:

La estructura jerárquica a utilizar, es la siguiente:

Data Type Tree of Leaves and Nodes (tree(leaves,nodes), tr)

operators

·leaf	leaves X name	-> tree
·node	nodes X name	-> tree
·insprogeny	tree X tree	-> tree
insnode	tree X nodes	-> tree
delprogeny	tree X name	-> tree
getname	tree	-> name
getleaf	tree	-> leaves
getnode	tree	-> nodes
getprogeny	tree X name	-> tree
isleaf	tree	-> boolean

end

Un objeto gráfico jerárquico se define entonces como:

Data Type Picture-Tree (pictree, pt)

enrichment

pictree ≡ tree (spicture, grtransf)

extension

*display pictree -> picture

end

Esta operación de display define la visualización del objeto gráfico.

Los objetos elementales de los cuales están compuestos los pictrees, pueden ser curvas, superficies paramétricas o estructuras más simples que representen las componentes del objeto.

```
operations
    ·end                                          -> trajectory
   *·inscontrol     trajectory X item X knot      -> trajectory
     addcontrol     trajectory X item X knot      -> trajectory
     i_knot         trajectory X integer          -> knot
     getcontrol     trajectory X knot             -> item
    ·start          trajectory                    -> knot
     finish         trajectory                    -> knot
     interpolation  trajectory X knot             -> item
end
```

La operación de interpolation define el tipo de la interpolante, es por ello que sus axiomas serán definidos en cada enrequecimiento particular.

Si los objetos jerarquicos estan compuestos por curvas parametricas:

```
Data Type Simple Picture of Parametric Curve (spicture, sp)

enrichment
     spicture = trajectory (point, real)
extension
     *display  spicture              -> picture
end
```

Por ultimo:

```
Data Type Point (point, p)

operations
    ·point      real X real X real          -> point
     origin                                 -> point
     x_axis     point                       -> real
     y_axis     point                       -> real
     z_axis     point                       -> real
     sum        point X point               -> point
     diference  point X point               -> point
     distance   point X point               -> real
end
```

DESCRIPCION INFORMAL DEL MODULO DE ANIMACION

El proceso de descripción de la animación se ha dividido en varias partes:

- Definición de los grados de libertad y restricciones de los objetos a animar (creación de los actores)
- Definición de movimientos sobre sistemas articulados (creación del guión)

- Designación de actores para el guión (película)

Notese que las dos primeras etapas del proceso de animación son independientes, con lo cual es posible tener bibliotecas de actores y guiones, los cuales pueden asignarse con gran libertad.

Definición de los Actores

Los actores son objetos jerárquicos a los cuales se han asociado articulaciones en algunos de sus niveles de jerarquía.

Estas articulaciones pueden ser de dos tipos:

- Rotacionales:
  Los movimientos posibles consisten de rotaciones con respecto al sistema de coordenadas asociado a la articulación. La posición, nos indica los ángulos de rotación en los distintos ejes.

- Traslacionales:
  Los movimientos posibles son traslaciones de la articulación.

Definición del Guión

Los movimientos de una articulación se definen mediante interpolación. Se basan en los trabajos de Bartels y Kochanek [KoBa84] referentes al uso de parámetros de continuidad, bias y tensión sobre interpolantes Hermite con el fin de obtener mayor expresividad temporal en la curva interpolada.

Un script, será entonces una estructura jerárquica de movimientos, los cuales indican el valos de las articulaciones en cada instante de tiempo.

Asignación del Script al Actor

La asignación de un Script a un Actor, debe tomar en cuenta tanto las restricciones que el actor posee, como la estructura jerárquica del mismo. Esto es:

- La jerarquía del Actor debe ser compatible con la del Script.

- Ningún movimiento definido en el Script debe violar las restricciones que posee la articulación.

Debemos definir el significado de compatibilidad utilizado aqui:

Sea A el conjunto de articulaciones presentes en el Actor, y $(A,<=)$ el conjunto parcialmente ordenado definido por la jerarquía de las articulaciones.

Sea M el conjunto de movimientos presentes en el Script, y $(M,<<=)$ el conjunto parcialmente ordenado definido por la jerarquía de los movimientos.

Sea D una función de M en A que asigna el Script al Actor.

$$F: M \rightarrow A$$

Llamemos M_D al dominio de F, M_D subconjunto de M.
Llamemos A_D a la imagen de M_D bajo F, A_D subcunjunto de A.

Definimos que la designación del Script es compatible con el Actor, si:

$(A_D,<=)$ es un CPO, $(M_D,<<=)$ es un CPO y F define un isomorfismo de $(M_D,<<=)$ en $(A_D,<=)$.

Especificación del Actor

Veamos algunas de las firmas de los tipos componentes de Actor:

Data Type Articulation of Actors (articulation, ar)

operations

·rigid		-> articulation
·traslartic		-> articulation
·rotatartic		-> articulation
·insrestriction	articulation X restriction X name	-> articulation
getrestriction	articulation X name	-> restriction
traslation	articulation	-> point
rotation	articulation	-> rotator
istraslrestricted	articulation X point	-> boolean
isrotatrestricted	articulation X rotator	-> boolean
setrotat	articulation X rotator	-> articulation
settrasl	articulation X point	-> articulation

end

Las restricciones serán expresadas como curvas que definen la frontera de un volumen donde es puede mover la articulación.

Data Type Restriction of Articulation (restriction, rt)

enrichment

restriction ≡ trajectory (point, real)

extentions

frontier restriction -> region

end

La operación frontier nos permite conocer la región que define el la superficie de la interpolante, y de esta forma podremos verificar si la restricción se respeta o no, al intersectarla con la trayectoria del movimiento a verificar.

Data Type Rotator (rotator, rt)

operations

·rot	real X real X real	-> rotator
origin		-> rotator
xy_angle	rotator	-> real
z_xyangle	rotator	-> real
spin	rotator	-> real
sum	rotator X rotator	-> rotator
diference	rotator X rotator	-> rotator
transfor	rotator	-> geomfcn
pointof	rotator	-> point

end

La operacion **transfor** es la encargada de indicar cual es la transformación geométrica asociada a la posición actual de la articulación. **pointof** efectua una transformación del espacio esferico en que se expresan los **rotators** al espacio cartesiano de los puntos.

Una vez especificados estos tipos, podemos definir a un actor como un Tipo Abstracto que posee la siguientes características.

Data Type Tree Structured Actor (actor, ac)

enrichment

actor ≡ tree (pictree, pair (articulation, grtransf))

extensions

actor	pictree	-> actor
addart	actor X articulation X name	-> actor
getartic	actor	-> articulation
gettransf	actor	-> grtransf
setartic	actor X articulation	-> actor
getpos	actor	-> grtransf
isrigid	actor	-> boolean
*display	actor	-> picture

end

En el proceso de construcción del actor pueden existir nodos de la figura que no poseen ninguna articulación, aunque son ancestros de nodos que contienen articulaciones. En estos casos dichos nodos tendrán una

articulación rígida. Notese que bajo el concepto de actor, se puede incluir a la escena que se esta animando, formando cada uno de los diversos objetos que en ella se mueven, una parte de un objeto mayor.

Especificacion del Script

La base del Script son los movimientos, veamos como estan definidos:

```
Data Type Movement of Articulations (movement, mm)

operations
     *rotmov     animation (rotator) -> movement
     *traslmov   animation (point)   -> movement
      istrasl    movement            -> boolean
      start      movement            -> time
      finish     movement            -> time
end

Data Type Animation Script (script, sr)

enrichment
      script = tree (nulltype, pair(time,movement))
extension
      attime     script X time       -> script
      insmov     script X movement   -> script
      entry      script              -> time
      move       script              -> movement
end
```

Como se puede notar los movimientos colocados en el Script se efectuarán en el instante de tiempo especificado, relativo al inicio de dicho movimiento en el script; la operación attime y entry permiten colocar e inspeccionar, respectivamente, el tiempo de inicio del movimiento.

Especificación de la película

```
Data Type Movie (movie, mv)

enrichment
      movie = tree (actor, triple (time, movement, grtransf))
```

```
extension
     movie      script X actor       -> movie
     addmovie   script X movie       -> movie
     iscompat   script X actor       -> boolean
     start      movie                -> time
     finish     movie                -> time
     ontime     movie X time         -> actor
    *display    movie X time         -> picture
end
```

El proceso de animación se puede ver como la evaluación de la operación de display cada $^1/_{30}$ de segundo.

ASPECTOS DE LA ESPECIFICACION AUN POR RESOLVER

Hasta el momento, hemos logrado especificar los tipos que componen el Módulo de Animación, faltan por tratar los aspectos referentes a la interface con el usuario, en particular:

Especificación del proceso de construción de los actores, movimientos, y script.

Especificación del lenguaje de control global de la película; es necesario poder realizar cambios de alto nivel sin tener que modificar los objetos que componen la película. (Lenguaje de Coordinación. Ej.: DIAL [FeSB82])

CONCLUSIONES

Se ha podido comprobar que la metodología de especificación propuesta por Mallgren permite especificar gran parte de los aspectos referentes al proceso de definición de imagenes tridimensionales (en su trabajo sólo analiza el problema 2-D).

El uso de tipos abstractos en la especificación del sistema ha permitido una definición clara y modular. Un cambio importante, por ejemplo, el tipo de objetos jerárquicos que se tienen o la forma en que las restricciones son especificadas, conlleva solamente el cambio de ciertos operadores dentro de la especificación que hemos realizado. De la misma

forma, un cambio en la manera en que los movimientos son especificados, se traduce en la modificación de pocas operaciones.

BIBLIOGRAFIA

[FeSB82] Feiner, S., Salesin, D., and Banchoff, T. "DIAL: A Diagramatic Animation Language", IEEE Computer Graphics & Animation, September 1982.

[Gogu78] Goguen, J.A., Thatcher, J.W., and Wagner, E.G. "An initial algebra approach to the specification, correctness and implementation of abstract data types, in Yeh", R.T. (ed) Current Trends in Programming Methodology, vol. IV, Data Structuring, Prentice Hall, Englewood Cliffs, New Jersey, 1978.

[Gutt75] Guttag, J.V. "The specification and application to programming of abstract data types", Tech. Report CSRG-59, University of Toronto, Toronto, Canada, September 1975.

[KBad82] Korein, J.U., and Badler, N.I. "Techniques for Generating the Goal-Directed Motion of Articulated Structures", IEEE Computer Graphics & Animation, September 1982.

[KoBa84] Kochanek, D., Bartels, R. "Interpolating Splines with Local Tension, Continuity, and Bias Control", Procedings Siggraph'84 July 84, 18(3), 33-41.

[Mall82] Mallgren, W. "Formal Specification of Interactive Graphics Programming Languages". ACM Distinguished Doctoral Dissertation Series, The MIT Press 1982.

[Zill74] Zilles, S.N. "Algebraic specification of data types, Project MAC Progress Report 11 (1974), Massachusetts Institute of Technology, Cambridge, Mass.,
52-58.