

VI Anexo (documento técnico)

“*In Silico*”: sistema computacional de vida artificial

1. Antecedentes

In Silico es un ambiente de simulación visual tridimensional, en donde varios objetos crecen en función de sus características e interacciones.

El patrón de crecimiento de cada objeto es modelado a través de sistemas L.

2. Desarrollo

a. Organismos

15.<

Los Sistemas L son una formalización matemática de la teoría axiomática del desarrollo biológico propuesta por el biólogo Aristid Lindenmayer en 1968. Recientemente los sistemas L han encontrado muchas aplicaciones en la computación gráfica. Las dos principales áreas incluyen generación de fractales y modelos realistas de vegetales.

Un punto central de los sistemas L es la noción de reescritura, tal y como sucede con las gramáticas regulares o gramáticas libres de contexto. La idea básica es sustituir símbolos predecesores con secuencias de símbolos, el patrón de sustitución está determinado por un conjunto de reglas de sustitución o producciones. La reiteración finita de las producciones de un sistema L produce palabras. El proceso de reescritura puede realizarse de manera recursiva.>

Los sistemas que operan bajo cadenas de caracteres han sido los más estudiados. En 1957 Chomsky aplica los sistemas formales de la teoría general de sistemas ^[g] al estudio de los lenguajes. Propuso un modelo capaz de representar la sintaxis de cualquier lenguaje, que llamó la atención de los computólogos interesados en aplicar los sistemas recursivos y gramáticas formales para representar el crecimiento.

Lindenmayer introdujo un nuevo tipo de gramática que posteriormente tomó el nombre de sistema L. La diferencia esencial con la gramática de Chomsky se encuentra en la manera de aplicar las producciones. En el sistema de Chomsky, las producciones son aplicadas secuencialmente, mientras que en el sistema L estas son aplicadas en paralelo, reemplazando simultáneamente todos los símbolos de una sola palabra.

12.<

Las palabras o expresiones pertenecientes al lenguaje generado por un sistema L tienen una interpretación geométrica en el espacio, esto es, cada expresión derivada a partir de un sistema L se corresponde con un objeto que puede ser situado en el espacio.

El tipo de sistema L utilizado en In silico es un sistema L paramétrico- estocástico.

Los símbolos paramétricos son símbolos de un alfabeto asociados con números reales.

Además de expresiones paramétricas que resultan en estructuras cambiantes en el espacio, estos sistemas L asignan una probabilidad de sustitución a cada producción, haciendo que el lenguaje generado sea "inestable". >

14.<

El alfabeto está conformado por símbolos de dos categorías: símbolos no terminales y terminales, esto es, símbolos sustituibles y símbolos no sustituibles. Comúnmente, los símbolos terminales tienen una interpretación geométrica, mientras que los no terminales sólo sirven para definir un patrón de desarrollo a través de una sustitución.

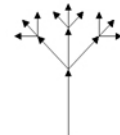
Tomemos un ejemplo contenido en *The Algorithmic Beauty of Plants*:

Sean

Axioma = X

Producción 1 = X -> A [+X][X][-X]

Producción 2 = A -> FF



En este caso X se reemplazara por una fórmula que significa que la rama se divide en 3 ramas.

“ F ” significa trazar una línea. (“ FF ” una línea después de la primera)

“ [] ” significa que efectúe la acción y regrese a su punto de origen

“ + ” y “ - ” estipulan una rotación de la línea con respecto a la orientación de origen

Esto se interpreta de la manera siguiente: el axioma es el modulo básico, en cada generación, busca si alguno de los elementos que conforman la expresión debe sustituirse. Entonces encuentra la Producción 1 y reemplaza la X por la fórmula correspondiente, posteriormente ejecuta la instrucción de crecimiento. Al tiempo que va leyendo las letras va reemplazándolas si es el caso, y si no continúa hasta el término de la expresión. >

La diferencia entre la gramática de Chomsky y la de Lindenmayer refleja la motivación biológica de los sistemas L. Las producciones intentan capturar la división celular en los organismos multicelulares donde las divisiones se dan de manera simultánea. El sistema L permite no sólo la descripción y reconstrucción de estructuras simétricas bidimensionales, sino también tridimensionales, formas celulares esféricas, crecimientos caóticos, interrumpidos o influenciados por fuerzas físicas.

El código que se incorporó al código de agentes se llama “Turtle” en alusión a los primeros códigos en lenguaje “Logo”. En este lenguaje una tortuga en pantalla dibujaba líneas o ángulos en función de los parámetros recibidos. Cada símbolo (“F”, “+”, “-”, “&”) corresponde a una instrucción. Cada una (*move*, *turn*, *set*) se acompaña del eje en el que debe actuar (*forward*, *turn*, *pitch*, *roll*), además entre paréntesis se agrega el parámetro de magnitud. Estos valores se le asignan a la tortuga “*turtle*”, que a su vez es un agente, y ésta dibuja en el espacio tridimensional el organismo descrito.

13.<

```
'F': turtle.moveForward (param);
'f': turtle.moveForward (param,false);
'+': turtle.turnAroundUp (param);
'-': turtle.turnAroundUp (-param);
'&': turtle.pitchAroundLeft (param);
'^': turtle.pitchAroundLeft (-param);
'/': turtle.rollAroundHeading (param);
'\\': turtle.rollAroundHeading(-param)
'|': turtle.turnAroundUp (Math.PI);
 '[': turtle.pushState ();
 ']': turtle.popState ();
 '#': turtle.setWidth (param);
 '!': turtle.setWidth(param);>
```

b. Ambiente

10.<

Los sistemas de Inteligencia Artificial (IA) ponen en ejecución conocimientos de sentido común o especializados y los explotan en mecanismos de raciocinio, es decir, en sistemas deductivos a base de conocimiento. El fundamento de esta aproximación de IA es la posibilidad de representar el saber en diferentes tipos de lógica.

Otro aspecto de la inteligencia a base de conocimientos es el de la inteligencia colectiva. Esta consiste en suscitar un comportamiento inteligente de manera emergente por cooperación entre un conjunto de entidades (agentes) con autonomía y tareas propias. Estos agentes pueden ser dotados de una capacidad cognoscitiva, de conocimientos previos y de mecanismos de raciocinio. En este caso hablamos de "sistema multiagentes". >

En los sistemas multiagentes cada agente tiene un comportamiento reactivo y está desprovisto de inteligencia. Es de la interacción entre agentes que emerge un comportamiento inteligente.

11.<

El sistema multiagentes en *In silico*

La interacción entre objetos y la base de tiempo utilizada para permitir el desarrollo de los mismos es controlada por una colección de agentes. Los agentes utilizados son del tipo reactivo, por lo que el comportamiento del sistema en su conjunto es resultado de la comunicación entre estos.

Se definen 5 tipos de agentes según sus funciones: un agente controlador del *display*; un agente maestro que determina la cantidad y los tipos de objetos en la simulación, así como también el momento en que se desarrollan; un agente que sincroniza el audio con el *display*; un agente que controla el tiempo, un agente que genera las expresiones paramétricas de un sistema L y un agente que las interpreta y convierte en geometría.

Al inicio de la simulación se crean un controlador de *display*; un sincronizador de audio y un agente maestro que crea a su vez una cantidad determinada de agentes generadores, intérpretes y de tiempo en la misma proporción, esto posibilita que cada objeto se desarrolle independientemente de los demás.

El desarrollo de un objeto pasa por cuatro fases: crecimiento, reproducción, envejecimiento y muerte. Los eventos de crecimiento, envejecimiento y muerte, dependen, aunque no exclusivamente, del tiempo, mientras que el evento de reproducción depende de la interacción con otros objetos en el ambiente. El tiempo de vida de un objeto puede verse reducido por la cantidad de veces que ocurre una interacción. >

Se dice que ocurre una interacción cuando un objeto invade el espacio vital de otro objeto, definido como un volumen alrededor del objeto en cuestión.

Es a través de la comunicación inter-agente que los objetos pueden reaccionar, ésta se lleva a cabo mediante el envío y recepción de mensajes asíncronos, esto es, que no se requiere el acuse de recibo de éstos, lo cual hace posible una interacción mucho más ágil.

3. Objetivo

- A. Los objetos deben simular un sistema vivo a través de su comportamiento individual y colectivo.
- B. El sistema debe simular el crecimiento orgánico mediante la generación de objetos 3D en tiempo real:
 - Configurar y crear los diferentes objetos en crecimiento.
 - Arbitrar el crecimiento de los diferentes objetos.
 - Determinar el crecimiento de cada objeto.
 - Establecer una base de tiempo para cada objeto.
 - Especificar el desarrollo de cada objeto.
 - Administrar el *display*.
 - Interpretar comandos para representación tridimensional.
 - Refrescar el *display*.
 - Atender las acciones del usuario.
 - Sincronizar el *display* con el nodo de audio.

Descripción general

El sistema es un entorno gráfico tridimensional por computadora a través del cual, el espectador podrá ver el desarrollo en tiempo real de una colección de objetos.

El sistema se ejecuta en una sola máquina.

El desarrollo de un objeto en el sistema implica su orientación espacial, su multiplicación o eliminación condicionada por variables del entorno, como la proximidad con algún otro objeto. Todo lo anterior es controlado por algoritmos recursivos y de vida artificial.

El desarrollo del ambiente depende fuertemente del estado inicial que se define como la colección de valores que tienen las variables reguladoras en el momento de iniciar la ejecución del sistema.

Al inicio de la ejecución del sistema se crean el agente responsable de crear los agentes de los objetos, el agente de *display* y el agente de audio. Al terminar la creación de estos tres, los activa dando inicio a la simulación.

Estado inicial

El estado inicial está conformado por los valores de las siguientes variables reguladoras al momento de iniciar la ejecución del sistema:

- Dimensiones del universo.
Valores x,y,z que determinan el tamaño del universo a simular. Definen un paralelepípedo que condiciona las dimensiones de los demás objetos.

- Tipo de objetos primitivos.
Un objeto primitivo es la geometría de base que se utiliza para crear un objeto en el sistema. La geometría que define a un objeto primitivo debe poseer la siguiente característica:

Un objeto primitivo debe tener por lo menos dos puntos, que pueden o no ser parte de su geometría, ubicados de tal forma que al concatenar espacialmente otro objeto primitivo en uno de esos puntos, las geometrías de ambos objetos queden unidas en una.

El objeto primitivo más simple es una superficie plana triangular.

- Cantidad de objetos origen
Un objeto origen es una geometría resultado de la concatenación espacial de objetos primitivos.
- Configuración de los objetos origen
La configuración de un objeto origen es el resultado de varias concatenaciones espaciales de objetos primitivos.
- Disposición espacial de objetos origen
La disposición espacial de los objetos origen se refiere a la localización en el universo de los objetos origen así como su orientación.
- Velocidad de crecimiento
Es la tasa a la que se realizan los cálculos de orientación espacial de la siguiente generación de objetos primitivos.
- Condiciones del ambiente
Están determinadas a su vez por una colección de variables, las cuales son:
 - Fuentes de luz:
 - Tipos de fuentes de luz: ambiental, direccional, puntual, tipo *spot*
 - Cantidad de fuentes de luz
 - Localización y orientación de las fuentes de luz
 - Obstáculos.

- Cantidad de obstáculos
 - Tipos de obstáculos: obstáculo esférico y obstáculo cúbico
- Son cuerpos que ofrecen resistencia a la penetración por parte de otros objetos: La región que ofrece resistencia está determinada por la forma, el radio y la influencia y el factor de penetración.
- Localización, orientación y apariencia de obstáculos
- Ubicación en el espacio, hacerlos visibles o invisibles.
- Atractores.
 - Tipos de atractores: atractor esférico

Son cuerpos que atraen los apéndices de los objetos

 - Cantidad de atractores
 - Localización, orientación y apariencia de los atractores

La ubicación y arreglo espaciales de los atractores. También es posible hacer visibles o invisibles a los atractores.

 - Posición y orientación del observador

Es la ubicación desde la cual se calcula la imagen que ha de ser transferida a la pantalla o al proyector.

Características de los usuarios.

El espectador es aquella persona que presencia el proceso de desarrollo de los objetos en el sistema. Para él, el sistema es autónomo.

El operador es aquella persona que propone el estado inicial. Es la persona que decide, en cualquier momento, iniciar o detener el sistema.

Restricciones.

Una vez iniciado el sistema, el operador no puede alterar ninguna de las variables que constituyen el estado inicial. Se deberá detener, alterar los valores de las variables y volver a iniciar la ejecución.

4. Características técnicas del sistema. (por Ing. Marco Antonio Montes de Oca)

Escenarios.

La segunda etapa es identificar escenarios de operación del sistema con el fin de determinar la secuencia de operaciones que se ejecutarán desde una situación determinada.

Escenario 1. Inicio del sistema.

Descripción: Al inicio de la ejecución del sistema se crea un agente que tendrá la responsabilidad de crear los agentes que controlaran el crecimiento y desarrollo de un objeto del mundo simulado. También será responsable de crear el agente administrador de *display* y el agente sincronizador de eventos de audio. Al terminar la creación de todos los agentes necesarios procederá a su activación, aquí es en donde en realidad iniciará la simulación. (fig. 1)

Escenario 2. El agente controlador de objeto es iniciado por el agente maestro.

Descripción: El agente maestro envía a cada agente controlador de objeto un mensaje de activación. Estos al recibir el mensaje, crean un agente base de tiempo, y un agente generador de comandos de crecimiento. El agente base de tiempo es independiente y determinará de manera autónoma el momento en el cual el objeto se desarrollará potencialmente.

(fig. 1) No existe mayor detalle pues no existen mensajes entre agentes, sólo son creados.

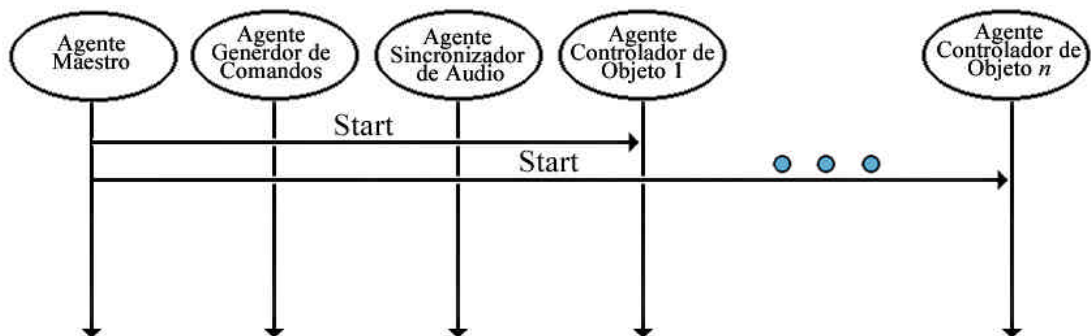


Fig 1. Diagrama de secuencia para el escenario 1 y 2.

Escenario 3. El agente base de tiempo determina que el objeto que controla debe crecer. El sistema está bloqueado.

Descripción: Transcurrido el tiempo que el agente base de tiempo determina, éste envía un mensaje al agente generador de comandos correspondiente al agente controlador de objeto que lo creó y al mismo tiempo disemina el mismo mensaje para que el agente sincronizador de audio lo atienda y envíe un paquete UDP al nodo controlador del audio. A su vez, el agente generador de comandos solicita el bloqueo del sistema al agente maestro para poder efectivamente desarrollarse. Si el sistema se encuentra bloqueado, el agente maestro le regresa un mensaje de espera, en cuyo caso el agente generador de comandos espera un tiempo preestablecido antes de volver a intentar bloquear el sistema.

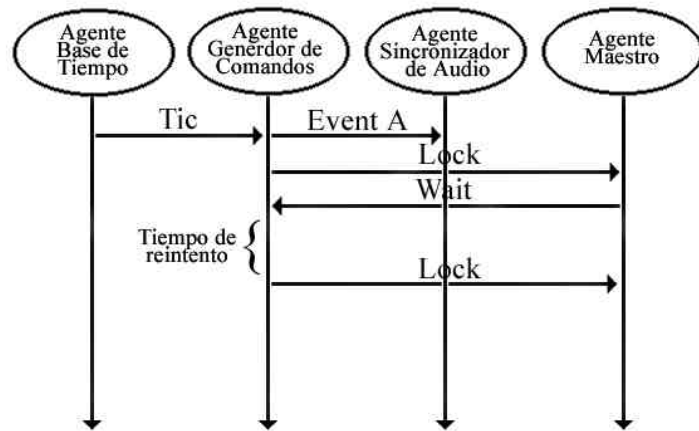


Fig. 2. Diagrama de secuencia del escenario 3.

Escenario 4. El agente base de tiempo determina que el objeto que controla debe crecer. El sistema no se encuentra bloqueado, el objeto está en etapa de crecimiento y al agregarse al mundo simulado, no interacciona con ningún otro objeto.

Descripción: Transcurrido el tiempo que el agente base de tiempo determina, éste envía un mensaje al agente generador de comandos correspondiente al agente controlador de objeto que lo creó y al mismo tiempo disemina el mismo mensaje para que el agente sincronizador de audio lo atienda y envíe un paquete UDP al nodo controlador del audio. A su vez, el agente generador de comandos solicita el bloqueo del sistema al agente maestro para poder efectivamente desarrollarse. Dado que el sistema no se encuentra bloqueado, el agente maestro le regresa un mensaje de autorización, en cuyo caso el agente generador de comandos determina el siguiente estado al que debe pasar el objeto que controla y envía el mensaje correspondiente al agente sincronizador de audio. Una vez calculado el siguiente estado, el agente generador envía la secuencia de comandos al agente controlador de objeto para su interpretación, esto es, para generar la geometría que lo representará en el espacio. Justo después de ello, el agente generador de comandos envía al agente maestro un mensaje para desbloquear el sistema. Paralelamente, el agente controlador de objeto, después de interpretar la secuencia de comandos, envía un mensaje de refresco de *display* al agente administrador de *display*. (fig. 3)

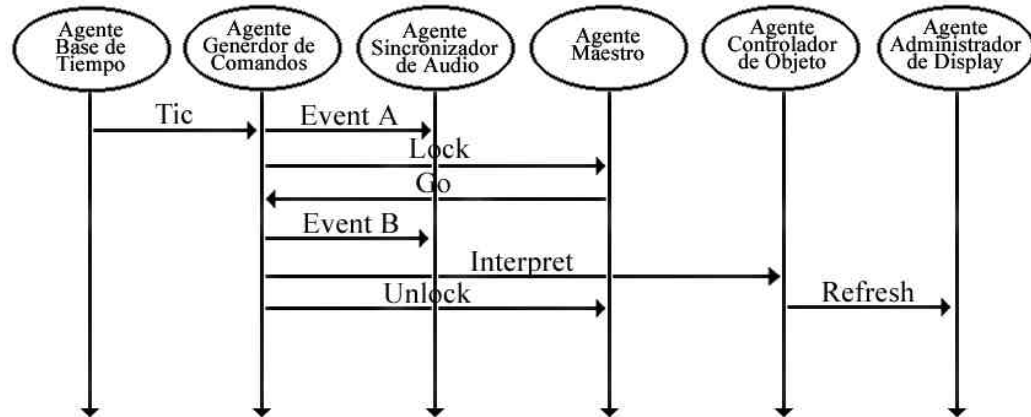


Fig. 3. Diagrama de secuencia del escenario 4.

Escenario 5. El agente base de tiempo determina que el objeto que controla debe crecer. El sistema no se encuentra bloqueado, el objeto está en etapa de crecimiento y al agregarse al mundo simulado, interactúa con otro objeto.

Descripción: Transcurrido el tiempo que el agente base de tiempo determina, éste envía un mensaje al agente generador de comandos correspondiente al agente controlador de objeto que lo creó y al mismo tiempo disemina el mismo mensaje para que el agente sincronizador de audio lo atienda y envíe un paquete UDP al nodo controlador del audio. A su vez, el agente generador de comandos solicita el bloqueo del sistema al agente maestro para poder efectivamente desarrollarse. Dado que el sistema no se encuentra bloqueado, el agente maestro le regresa un mensaje de autorización, en cuyo caso el agente generador de comandos determina el siguiente estado al que debe pasar el objeto que controla. Una vez calculado el siguiente estado, el agente generador envía la secuencia de comandos al agente controlador de objeto para su interpretación, esto es, para generar la geometría que lo representará en el espacio. Justo después de ello, el agente generador de comandos envía al agente maestro un mensaje para desbloquear el sistema. Paralelamente, el agente controlador de objeto, después de interpretar la secuencia de comandos, envía un mensaje de refresco de *display* al agente administrador de *display* que detecta que el nuevo objeto colisiona con otro objeto, enviando un mensaje al agente controlador de objetos. El agente controlador de objeto determina si es sólo una interacción común o se trata de una interacción de réplica o de envejecimiento. Envía el evento correspondiente al agente sincronizador de audio. (fig. 4)

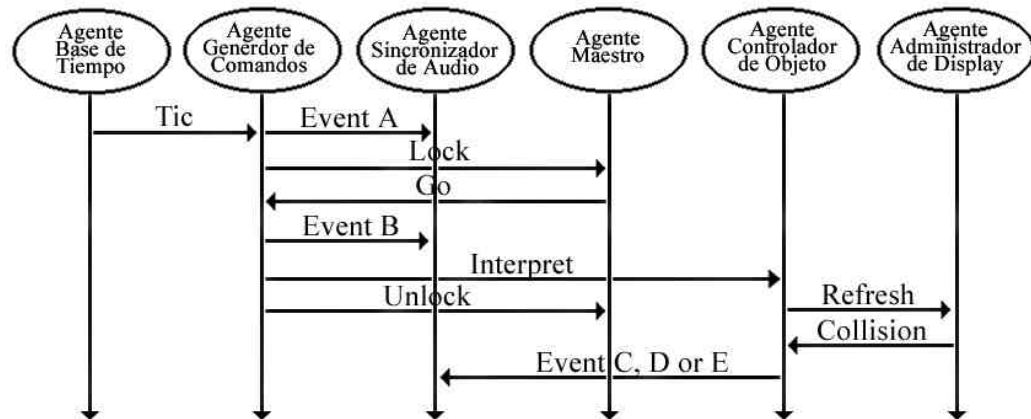


Fig 4. Diagrama de secuencia para el escenario 5.

Escenario 6. El agente base de tiempo determina que el objeto que controla debe crecer. El sistema no se encuentra bloqueado, el objeto está en etapa de crecimiento y al agregarse al mundo simulado, interactúa con otro objeto provocando su propia muerte.

Descripción: Transcurrido el tiempo que el agente base de tiempo determina, éste envía un mensaje al agente generador de comandos correspondiente al agente controlador de objeto que lo creó y al mismo tiempo disemina el mismo mensaje para que el agente sincronizador de audio lo atienda y envíe un paquete UDP al nodo controlador del audio. A su vez, el agente generador de comandos solicita el bloqueo del sistema al agente maestro para poder efectivamente desarrollarse. Dado que el sistema no se encuentra bloqueado, el agente maestro le regresa un mensaje de autorización, en cuyo caso el agente generador de comandos determina el siguiente estado al que debe pasar el objeto que controla. Una vez calculado el siguiente estado, el agente generador envía la secuencia de comandos al agente controlador de objeto para su interpretación, esto es, para generar la geometría que lo representará en el espacio. Justo después de ello, el agente generador de comandos envía al agente maestro un mensaje para desbloquear el sistema. Paralelamente, el agente controlador de objeto, después de interpretar la secuencia de comandos, envía un mensaje de refresco de *display* al agente administrador de *display* que detecta que el nuevo objeto colisiona con otro objeto y lo ha hecho por una cantidad determinada de veces, después se inicia el envejecimiento, por lo que envía un mensaje al agente controlador de objetos. Este envía el evento correspondiente al agente sincronizador de audio. (fig. 5)

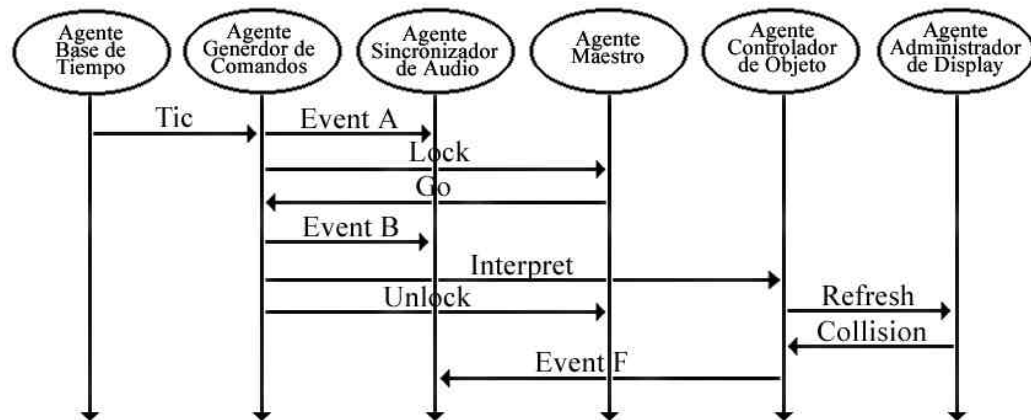


Fig 5. Diagrama de secuencia para el escenario 6.