

ESPECIFICACION DE REPORTES MULTIMEDIA

Olivera Marilenis , Meier Andreas

Departamento de Computación y Tecnología de la Información

Universidad Simón Bolívar

Caracas 1080-A , Apartado Postal 89000

Email : {olivera, meier} @ usb.ve

RESUMEN

Los Reportes Multimedia (RM), han sido propuestos como una redefinición al modelo tradicional de reportes. El contenido de un RM es el conjunto de datos multimedia, resultado de una consulta a una Base de Datos Multimedia. Estos datos son organizados y presentados en pantallas de despliegue, en las que el usuario puede activar la reproducción de sonidos, animación o enlaces entre conjuntos de datos, de manera interactiva. [Oliv 97]

La Generación de Reportes Multimedia (GRM) es el proceso de producción de estos reportes. Este proceso se inicia con el análisis de la consulta especificada a la base de datos multimedia y culmina con la presentación al usuario del RM. Una de las etapas intermedias en el proceso de generación es la especificación de los reportes, usando un lenguaje de especificación.

En este artículo se presenta detalladamente el uso de estándares de especificación hipermedia, para especificar los RM, así como la arquitectura de la herramienta diseñada para generar los reportes multimedia. Esta herramienta ha sido denominada Generador de Reportes Multimedia.

Palabras Claves : Multimedia, Hipermedia, Estándares de especificación, HyTime, Base de datos multimedia, Reportes Multimedia.

1. INTRODUCCION

Los Documentos Multimedia se conciben como presentaciones multimedia, interactivas no-lineales, en las que el usuario controla el ritmo y la trayectoria de la presentación [Vaug 93]. Los Paradigmas de Autoría son mecanismos de especificación de la estructura de un documento multimedia, donde se contemplan relaciones espaciales, temporales y semánticas, entre los datos de medios, y la interactividad asociada a éstos [Gibb 94]. Entre los paradigmas existentes se pueden señalar los lenguajes de programación, arquitectura de documentos multimedia estándar, el uso de herramientas de autoría y uso de gramáticas relacionales, entre otros.

El proceso de generación de reportes implica la selección de un paradigma de autoría que se adapte a las necesidades de generar automáticamente la especificación de los reportes multimedia. El Paradigma de Autoría seleccionado en esta investigación es "Arquitectura de Documentos Multimedia Estándar", que consiste en el uso de

un lenguaje de especificación hipermedia, para especificar los documentos multimedia [Bufo94a]. El lenguaje de especificación seleccionado es *HyTime* [ISO 92].

Los conceptos formales sobre tecnología hipermedia usados en esta investigación son los propuestos por [Bult 94] en el Modelo Hipermedia de Amsterdam (MHA) y por [Hala 94] en el Modelo Hipertextos de Dexter.

Como resultado de esta investigación, se han diseñado varios modelos de RM y la arquitectura de una herramienta para la generación de estos reportes. En este artículo se hace énfasis en la discusión de las dos últimas etapas del proceso de GRM, se presenta el uso de *HyTime* para la especificación de los RM y se discute la arquitectura de la herramienta diseñada.

En las proximas secciones de la introducción se describen brevemente las etapas iniciales del proceso de generación de reportes multimedia, las cuales han sido descritas ampliamente en [Oliv 97].

1.1 Proceso de generación de reportes multimedia

En este proceso se decide el modelo conceptual y el diseño real del reporte a generar, en base a un análisis de la consulta especificada por el usuario. En esta investigación se trabaja con la estructura genérica de una consulta a una base de datos usando SQL, y con conceptos del modelo relacional de datos [Codd 70]. Las etapas del proceso son:

- ⇒ Análisis de la estructura de la consulta.
- ⇒ Modelaje conceptual de los RM.
- ⇒ Asociación del modelo conceptual al diseño visual de los RM.
- ⇒ Especificación de los RM usando un lenguaje de especificación.
- ⇒ Presentación del reporte generado al usuario.

1.1.1. Análisis de la estructura de la consulta

En este análisis se determina el tipo de consulta, el número de campos de información requeridos y el tipo de medio de cada campo. Los parámetros usados en este análisis son : los argumentos de la cláusula de selección (SELECT) y la presencia de la cláusula de agrupamiento (GROUP BY), este parámetro determina la estructura simple u organizada en grupos de datos, de la tabla resultado (TR) de la consulta. Los resultados de este análisis son parámetros de entrada para las siguientes etapas del proceso.

1.1.2. Modelaje conceptual de los reportes

Consiste en una representación abstracta de los datos almacenados en TR y de las relaciones entre ellos. Los objetos lógicos que conforman el modelo conceptual son : datos atómicos , componentes simples , componentes compuestas, enlaces [Hala 94] y un documento definido como una colección de componentes relacionados. [Hard 94]

1.1.2.1 Modelos de reportes

Según la estructura de la TR se define el modelo conceptual del reporte. Se distinguen dos modelos :

1. Modelo Simple : asociado a una TR sin estructura. Se define por un conjunto de componentes simples, enlazadas entre si, en una componente compuesta.
2. Modelo Agrupado : asociado a una TR estructurada en grupos de filas. Se define por una componente índice, que contiene los campos de agrupamiento enlazados entre sí, a cada campo de agrupamiento se enlazan un conjunto de componentes simples, que contienen los datos asociados a éstos.

1.1.3.1 Asociación del modelo conceptual al diseño visual de los reportes :

En esta etapa, se obtiene la representación visual, de los RM. Esta representación se logra asociando un conjunto de elementos visuales al modelo conceptual del RM. Los elementos gráficos para la representación son : pantallas, botones, *scrollbars*, menús e iconos que representan tipos de datos. Por ejemplo, un ícono en forma de una nota musical para representar un dato tipo audio.

Las dos ultimas etapas del proceso de GRM : especificación de los RM usando HyTime y presentación del reporte generado al usuario, serán discutidas ampliamente a continuación.

2. ESTANDARES DE ESPECIFICACION HIPERMEDIA

Existen una variedad de estándares para estructuración de aplicaciones hipermedia. Entre ellos SGML, HyTime, MHEG, ODA, PREMO y Acrobat [Adie 93]. En esta sección se describirán las características relevantes de HyTime para la especificación de documentos hipermedia y se mostrará su aplicación para la especificación de los RM definidos inicialmente.

2.1 SGML

SGML - Lenguaje de Etiquetas Estándar Generalizado (ISO/IEC 8879) es un meta lenguaje, usado para definir lenguajes que permiten describir la estructura lógica de un documento de texto, en función de los elementos de información que éstos contienen.

Una definición de un lenguaje expresada en sintáxis SGML está contenida en una Definición de un Tipo de Documento (DTD). En éste se declaran los tipos de elementos que pueden existir en un documento, qué atributos puede tener cada uno de esos tipos, y como están relacionados jerárquicamente las instancias de estos tipos de elementos. Un DTD define una clase de documentos.

2.2 HYTIME

HyTime-Lenguaje de Estructuración Hipermedia basado en tiempo , es un subconjunto de SGML, el cual provee un medio, independiente de la aplicación, de especificar formato, estructura y direccionamiento en documentos hipermedia. HyTime es un estándar de la ISO (ISO/IEC 10744) [ISO92], y una aplicación de SGML ISO-8879.

Es usado como un modelo para definir un número arbitrario de tipos de documentos multimedia, donde cada tipo podría tener su propio DTD . HyTime especifica como los conceptos comunes a todos los documentos hipermedia pueden ser representados usando SGML. Estos conceptos incluyen :

- Asociación de objetos dentro de documentos, usando hiperenlaces.
- Posicionamiento y relación entre objetos en espacio y tiempo.
- Estructuración lógica de documentos.
- Inclusión de cualquier tipo de datos en el documento.

HyTime es definido formalmente por un conjunto de reglas, para la expresión estandarizada de información hipermedia, denominadas "Formas Arquitectónicas". Está estructurado como un conjunto de módulos que son incluidos, según sea necesario, para una aplicación particular. Hytime cuenta con seis módulos. Cada módulo soporta un conjunto de Formas Arquitectónicas. El conjunto de formas arquitectónicas soportado por un módulo particular determina las facilidades provistas por ese módulo.

3. ESPECIFICACION DE REPORTES MULTIMEDIA USANDO HYTIME

En esta investigación, HyTime ha sido seleccionado como lenguaje de especificación de los RM pues satisface los requerimientos de especificación de las características básicas de los reportes, como por ejemplo, la inclusión de cualquier tipo de datos en el documento, lo cual permite la inclusión de datos de medios almacenados en bases de datos externas. La asociación de objetos en el documento, usando hiperenlaces, es otra de las características fundamentales que se deben satisfacer. Adicionalmente, el uso de un estándar, permite el intercambio de documentos entre aplicaciones y plataformas. A continuación se describen las características de los reportes que serán especificadas usando HyTime.

3.1 Esquema del DTD para representar los reportes multimedia

Características de los reportes que serán representadas por el DTD :

- El reporte, como la unidad que contiene todos los elementos a presentar al usuario.
- Referencias a los datos en la base de datos, se tratan como entidades externas, pueden ser de cualquier tipo : audio, imágenes, texto.

- Despliegue y organización de los datos en una pantalla. Manejo de coordenadas del espacio de la pantalla, para la ubicación de los datos.
- Enlaces entre pantallas del reporte.
- Botones con imágenes sobrepuestas. Estos botones se usan como enlaces a otra pantalla del reporte.
- Botones que al ser presionados, disparan la ejecución de un sonido.
- Se toman en cuenta los tipos de datos que pueden aparecer en una pantalla.

3.2 Tipos de elementos en el DTD y su funcionalidad

En esta sección se enumeran los tipos de elementos definidos, que conforman el DTD, para especificar los reportes. Para cada elemento declarado se describe la semántica de sus atributos, tal como será interpretada por la aplicación. El DTD es presentado en el apéndice de este artículo. Los tipos usados para clasificar los elementos están declarados en el catálogo de formas arquitectónicas y en la especificación de HyTime SGML. [Gold93]

MMREPORT : actúa como el objeto raíz para los documentos. Pertenece a la forma arquitectónica *HyDoc*, este tipo contiene todo los elementos del documento, y es la raíz de la jerarquía de elementos en el documento. El atributo HyTime docmdu, indica que el dominio de las unidades de medidas del documento es una unidad cuya longitud tiene una relación uno a uno con la unidad de medida estándar virspace. Una instancia de este elemento contiene una instancia de reportfcs, cero o más instancias de *dimspec*, cero o más instancias de butnlink, y una instancia de Ejes x, y (el número de instancias está expresado por el modelo de contenido, encerrado en paréntesis en la definición del elemento).

REPORTFCS : es del tipo *fcs*. Define un espacio de coordenadas finito, definido por los ejes x, y . Este contiene uno o más elementos tipo pantallasched, que representan instancias separadas de este espacio de coordenadas, con eventos posicionados dentro de ellos. El atributo axisdefs indica que los ejes de este *fcs* son, en orden, el eje x , eje y. Estos elementos (eje x,y) son definidos posteriormente en el DTD.

PANTALLASCHED : es del tipo *evsched*. Este localiza una colección de eventos dentro de una instancia de un espacio de coordenadas finito. En este caso los eventos son despliegues de imágenes y texto, y una instancia de pantallasched es el espacio de la pantalla que contiene estos despliegues. También se especifica un botón next, que hace que se despliegue la pantalla a la que está enlazado, este botón es de tipo idrefs, que indica que puede enlazarse a una lista de pantallas. Este contiene cero o más imágenes, cero o más textos y cero o más botones, todos del tipo *event*.

TEXTO : tipo *event*. Especifica un archivo de texto a ser desplegado en la pantalla. El atributo exspec se refiere a un par de *dimspec* ó especificaciones de dimensión. Estas dimensiones pueden transformarse a cualquier posición a lo largo de los ejes.

IMAGEN : tipo *event*. Especifica un archivo de imagen a ser desplegado en la pantalla. El atributo exspec se refiere a un par de *dimspec* ó especificaciones de dimensión. Estas dimensiones pueden transformarse a cualquier posición a lo largo de los ejes.

BOTON : tipo *event*. Especifica el área activa de un botón. Esta área puede tomar cualquier forma. El atributo *botonlink* identifica el elemento que contiene la información de enlace de este botón. El atributo *exspec* define la posición de la imagen que especifica el área del botón.

BOTONLINK : tipo *clink*. Está asociado con un botón particular, y especifica que ocurre cuando el botón es presionado. Si el botón es un botón de enlace, este elemento indica donde está el final del enlace (atributo *linkend*), y que ocurre cuando se ejecuta el enlace. El atributo *id* es necesario, así este *botonlink* puede ser referido por un elemento tipo botón, como su enlace (en botón la referencia se hace a través del atributo *botonlink*).

Ejes *X*, *Y* : se usan para definir posicionamiento de objetos en la pantalla. Son de tipo eje.

3. ARQUITECTURA DEL GENERADOR DE REPORTES MULTIMEDIA

En esta sección se describe el diseño de la arquitectura de la aplicación propuesta, para la generación de reportes. Las instancias de los reportes son codificadas en HyTime, y almacenadas en una base de datos, usando el modelo de datos de HyTime. [Hala 94] Este diseño describe la arquitectura de la aplicación y el proceso de generación de reportes. La aplicación está estructurada en tres capas :

3.1 Almacenamiento

Constituida por la base de datos de la aplicación, donde se almacena meta-información de los reportes multimedia (Ver figura 1, base de datos de almacenamiento representada en el cuadro inferior). Esta meta-información está constituida por el modelo conceptual del reporte multimedia, y las especificaciones de presentación. Esta información, asociada a un tipo de reporte particular, es generada en la capa de procesamiento.

Las especificaciones de presentación contienen el diseño de los documentos, donde se detalla la organización visual de los datos en las pantallas de los documentos y los mecanismos de navegación en estas pantallas. La información almacenada en esta capa es interpretada por la aplicación al momento de ejecución. En la figura 1 la flecha entre capa de procesamiento y base de datos de almacenamiento representa la transferencia de información a la base de datos.

3.2 Procesamiento

En esta capa se procesan los requerimientos de información, y se genera la instancia del DTD correspondiente al reporte multimedia, se realizan las siguientes actividades:

- 1- Análisis de la consulta.
- 2- Definición de la estructura del reporte y especificación de los atributos de la presentación.
- 3- Generación instancia DTD (SGML-HyTime) del reporte predefinido.
- 4- Procesamiento HyTime de la instancia DTD (Ver sección 3.2.1).

En la base de datos asociada a esta capa se almacena la definición del tipo de reporte (DTD), y la instancia del DTD generada por el procesamiento. La base de datos asociada a esta capa almacena los aspectos de especificación en HyTime y su codificación. En la figura 1 la flecha entre capa de procesamiento y base de datos de procesamiento representa la transferencia de información a la base de datos.

3.2.1 Modelo de procesamiento de documentos HyTime

Las aplicaciones de HyTime, usan el modelo de procesamiento de documentos, definido por HyTime y SGML. En la etapa inicial de este procesamiento, se usan dos herramientas : un Interpretador SGML y un Programa Agente HyTime. [Koeg 93] [ISO 92]

Este modelo de procesamiento genera información acerca de la estructura del documento, que es independiente de la aplicación y de la presentación. Esta información es almacenada en la base de datos asociada a la capa de procesamiento y usada posteriormente por la aplicación para determinar como va a ser presentado el documento.

3.3 Ejecución - Presentación

En esta capa se controla la captura de los requerimientos de información del usuario, a través de una herramienta visual para la construcción de consultas a base de datos. Se proveen herramientas para la presentación de los reportes multimedia e interacción del usuario con los reportes.

La figura 1 muestra el proceso de generación de reportes multimedia. En esta figura el primer cuadro (en dirección vertical) muestra la etapas de generación de reportes multimedia. En la etapa de generación de instancia de DTD, ésta es pasada como entrada al modelo de procesamiento de HyTime y la salida de este proceso la toma la aplicación de presentación, la cual se encarga de presentar de manera interactiva el reporte al usuario.

El segundo cuadro muestra una correspondencia entre las etapas del proceso de generación y la capa de la aplicación en la cual se ejecuta. En el cuadro inferior aparecen las bases de datos asociadas a las capas de almacenamiento y procesamiento, donde se almacena la información generada por el proceso de generación de reportes multimedia.

CONCLUSIONES

Los reportes multimedia : representan una forma dinámica, flexible y expresiva, de presentar la información multimedia almacenada en una base de datos. La GRM libera al autor de la carga de autoría del reporte multimedia.

El uso de estándares de especificación hipermedia permite el intercambio e interoperabilidad de los reportes multimedia entre plataformas y herramientas. SGML es la base de los lenguajes de etiquetas usados para la

especificación de documentos en internet, esta estandarización es beneficiosa en el sentido de no generar documentos propietarios a una aplicación aislada.

La arquitectura de una aplicación estructurada en capas, muestra el procesamiento de los reportes multimedia, usando el esquema de procesamiento propuesto por el estándar HyTime.

El DTD presentado en el apéndice es una muestra válida del uso de las características de HyTime como lenguaje de especificación hipertexto, para la especificación de los reportes multimedia

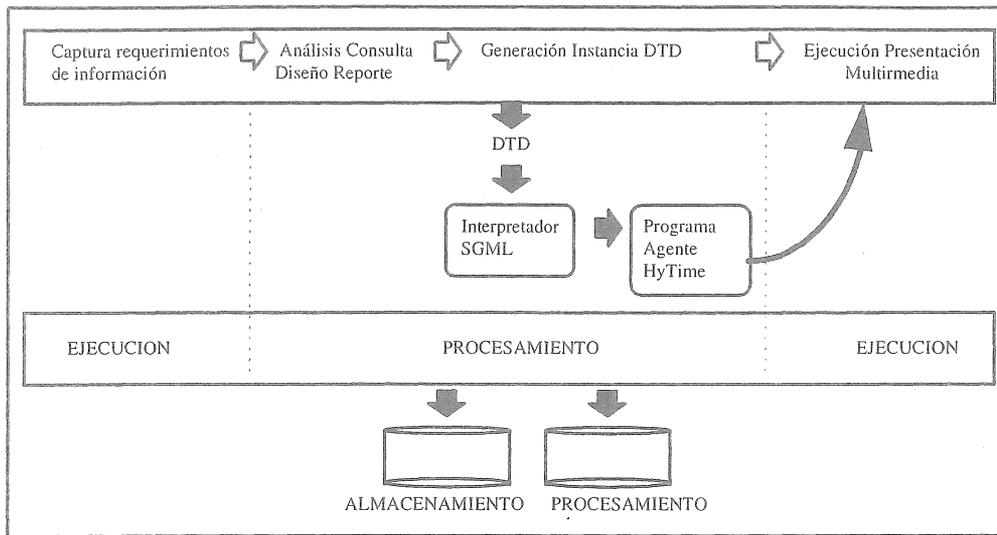


Figura 1 - Proceso de Generación de Reportes Multimedia

REFERENCIAS BIBLIOGRAFICAS

[Adie 93] Chris Adie, Network Access to Multimedia information, RARE Project OBR(93)015, Edinburgh University Computing Service, University Library Building, George Square-Edinburgh, EH8 9LJ, Great Britain. Second Edition 9 August 1993.

[Bufo94a] John F. Buford. "Uses of Multimedia Information", *Multimedia Systems*, Contributing Editor John Buford, ACM Press, SIGGRAPH Series, New York-NY, 1994, pp. 1-25.

[Bult94] Dick C. A. Bulterman, Lynda Hardman, Guido van Rossum, "The Amsterdam Hypermedia Model", in *Communications of the ACM*, February 1994 / Vol. 37, No. 2.

[Cood 70] E.F. Cood, A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM* 13, No. 6 (June 1970). Reprinted in *Communications of the ACM* 26, No. 1 (January 1983).

[Gibb 94] Simon J. Gibbs and Dionysios C. Tsihrizis. "Some Research Problems in Multimedia Programming", *Multimedia Programming*, ACM Press and Addison-Wesley, New York-NY, 1994.

[Gold 93] Charles F. Goldfarb, Catalog of HyTime Architectural Forms and HyTime SGML Specification, Version 2.0, June 28, 1993.

[Hala 94] Frank Halasz, Mayer Schwartz. The Dexter Hypertext Reference Model. K Grønbaek and R Trigg, Eds., *Communications of the ACM*, February 1994 / Vol. 37, No. 2.

[Hard 94] Lynda Hardman, Dick Bulterman, Guido van Rossum, "Structured Multimedia Authoring", In Proceedings of ACM Multimedia 93, Anaheim CA, 1993, pp 283-290.

[Hard 94] Lynda Hardman, Dick C.A. Bulterman, Guido van Rossum, "The Amsterdam Hypermedia Model", in Communications of the ACM, Febrero 1994 / Vol. 37, No. 2, pp. 51-62.

[Inte 92]International Standards Organization, Hypermedia/Time-Based Structuring Language (HyTime). ISO/IEC IS 10744, April 1992.

[ISO 92] ISO. ISO/IEC IS 10744, "Hypermedia / Time based Document Structuring Language (HyTime)", Abril 1992.

[Koeg 93] John F. Koegel, Lloyd W. Rutledge, John Rutledge, Can Keskin. HyOctane: A HyTime Engine for an MMIS, in Proceedings of ACM Multimedia 93, Anaheim CA, 1993, pp 129-136.

[Oliv 97] Olivera Marilenis, Meier Andreas, "The Generation of Multimedia Reports", aceptado para ser publicado en las actas de la conferencia "World Multiconference on systemics, cybernetics and informatics", a celebrarse en caracas, del 7-11 julio 1997.

[Vaugh 93] Tay Vaughan. "Multimedia Authoring Tools", *Multimedia Making it Work*, Osborne Mc Graw Hill, Berkeley-CA, 1993, pp. 217-264.

Apéndice

Reportes Multimedia - DTD

<!-- Realizado por : Marilenis Olivera.
Universidad Simón Bolívar, Caracas-Venezuela.

Especificación de una Tipo de Documento (DTD), usando HyTime, para representar los Reportes Multimedia, diseñados en este trabajo de investigación.

-->

<!-- MMREPORT (Reporte Multimedia) es el elemento que contiene todo el documento. Es de la forma arquitectónica de HyTime "doc". Este contiene un conjunto de pantallas que van a ser desplegadas en un orden determinado por la interacción del usuario. -->

<!element mmreport -o (reportfcs, dimspec*, butnlink*, x, y) >

<!attlist mmreport

-- Atributos HyTime para esta forma arquitectónica --

HyTime name #fixed HyDoc
docmdu cdata #fixed "virspace 1 1"

--- "virspace" es la unidad a la que se transforman todas las medidas en este documento -->

<!-- REPORTFCS (espacio de coordenadas finito del reporte) es el sistema de coordenadas según el cual son transformados las pantallas, imágenes y botones del reporte. Es de la forma arquitectónica "fcs" -->

<!element reportfcs -o (pantallasched+) >

<!attlist reportfcs

-- Atributos HyTime para esta forma arquitectónica --

HyTime name #fixed fcs
fcsmdu cdata #fixed "" -- Por defecto es igual a docmdu, virspace --

```
axisdefs cdata #fixed "x y" -- Se usan los ejes x , y -->
```

<!-- PANTALLASCHED (controlador de pantallas) contiene toda la información para una pantalla en particular. Es de la forma arquitectónica HyTime "evsched", como tal se encarga de controlar los objetos sobre la pantalla del reporte -->

```
<!element pantallasched -o (texto*, imagen*, boton*) >
```

```
<!attlist pantallasched
```

```
next idrefs #implied -- identificador genérico, usado en caso que este  
elemento sea punto de salida de enlaces, idrefs indica que pueden salir varios enlaces --
```

```
HyTime name #fixed evsched
```

```
id id #required -- identificador del elemento id --
```

```
>
```

<!-- TEXTO es la transformación de un texto particular de una pantalla del reporte, en un controlador de pantallas, o pantallasched. Es de la forma arquitectónica "event", y como tal especifica la localización en la cual va a ser desplegado el texto. Se indica el archivo de texto a desplegar y el formato. -->

```
<!element texto -o empty >
```

```
<!attlist boton
```

```
filename cdata #required -- archivo donde se localiza el texto --
```

```
format (ascii / postscript / ..) -- formato del archivo tipo texto --
```

```
HyTime name #fixed event
```

```
exspec cdata #required -- el dimspec, que especifica la dimensión de un  
evento, en este caso la localización de un texto a través de los ejes x,y -->
```

<!-- IMAGEN es la transformación de una imagen particular de una pantalla del reporte, en un controlador de pantallas, o pantallasched. Es de la forma arquitectónica "event", y como tal especifica la localización en la cual va a ser desplegada la imagen. Se indica el archivo de imagen a desplegar y el formato. Especificado para cubrir los reportes tipo detallado, donde aparecen las imágenes ampliadas, y no reaccionan al doble-click del ratón -->

```
<!element texto -o empty >
```

```
<!attlist boton
```

```
filename cdata #required -- archivo donde se localiza el texto --
```

```
format (gif / bmp / jpg / ..) -- formato del archivo de imagen --
```

```
HyTime name #fixed event
```

```
exspec cdata #required -- el dimspec, que especifica la dimensión de un  
evento, en este caso la localización de un texto a través de los ejes x,y -->
```

<!-- BOTON contiene la información de enlace para un botón de una pantalla. Es de la forma arquitectónica "event", y como tal éste especifica el área en la pantalla la cual responde a un doble-click del ratón -->

```
<!element boton -o empty >
```

```
<!attlist boton
```

```
filename cdata #required -- archivo donde se localiza la imagen que especifica el área del botón --
```

```
botonlink idref #implied -- punto de salida de enlace a otra pantalla --
```

```
HyTime name #fixed event
```

```
id id #required -- identificador del elemento id --
```

```
>
```

<!-- BOTONLINK especifica un enlace particular para un boton. Es de la forma arquitectónica "clink". Este es referenciado por una instancia de BOTON. El tipo del boton puede ser : "salir", al presionar del botón se termina la presentación, "enlace", al presionarlo se presenta la pantalla que esté referenciando a través de linkend y "ejecutar", al presionarlo se ejecuta el sonido asociado. La imagen será colocada sobre el botón y el sonido se ejecuta al presionar un botón -->

<!element botonlink -o empty >

<!attlist botonlink

type	(salir / enlace/ejecutar)	#required	-- tipo de acción --
filename	cdata	#required	--archivo donde se localiza la imagen que especifica el área del botón --
format	(gif / bmp/ jpg / ..)	#required	-- formato del archivo de imagen--
audfile	cdata	#required	-- archivo donde se localiza un sonido que puede ser ejecutado --
audformat	(wav / ..)	#required	-- formato del archivo de sonido--
HyTime	name	#fixed	clink
id	id	#required	-- declara el identificador del elemento, obligatorio, para usarse como punto de llegada de enlaces --
linkend	idref	#implied	-- identificador genérico, no obligatorio, que se usa en caso de que este elemento sea un punto de salida de enlaces. En este atributo se indica el valor del identificador del elemento fuente. >

<!--Ejes X , Y en el sistema de coordenadas finito. Pertenece a la forma arquitectónica "axis" -->

<!element x -o empty >

<!attlist x

HyTime	name	#fixed	axis
axismas	cdata	#fixed	"virspace"
			-- sistema de medidas usado (mdu) --
axismdu	cdata	#fixed	"" -- valor por defecto, igual a fcsmdu, virspace --
axisdim	cdata	#fixed	"1024" -- este eje tiene 1024 unidades, debe contener la longitud en pixels de una pantalla -->

<!element y -o empty >

<!attlist y

HyTime	name	#fixed	axis
axismas	cdata	#fixed	"virspace"
			-- sistema de medidas usado (mdu) --
axismdu	cdata	#fixed	"" -- valor por defecto, igual a fcsmdu, virspace --
axisdim	cdata	#fixed	"1024" -- este eje tiene 1024 unidades, debe contener la longitud en pixels de una pantalla -->