



**Escola Politècnica Superior
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Creació d'un sistema multipista de 4 pistes de so per al DSI Sound

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Sistemes de Telecomunicació

**AUTORS: Oriol Príncep Dols
Álvaro del Río Navarro**

DIRECTOR: Xavier Montero Amorós

SUPERVISOR: David Rincón Rivera

DATA: 5 de setembre de 2005

Títol: Creació d'un sistema multipista de 4 pistes de so per al DSI Sound

Autors: Oriol Príncep Dols
Álvaro del Río Navarro

Director: Xavier Montero Amorós

Data: 5 de setembre de 2005

Resum

La finalitat d'aquest programa és la creació de reportatges i cròniques radiofòniques com a projectes d'àudio, utilitzant diverses pistes de so, per a la seva posterior emissió. L'aplicació està enfocada a periodistes que treballin en una redacció de ràdio i que no tinguin necessàriament uns grans coneixements d'informàtica. El programa consta d'una interfície senzilla que permet treballar amb diverses pistes on s'hi poden afegir talls d'àudio prèviament enregistrats. Les seves característiques principals són:

- Possibilitat d'afegir, desplaçar, definir un interval o eliminar talls de so dins d'una pista.
- Mètode de Drag & Drop per afegir talls des de qualsevol localització.
- Reproducció d'una o diverses pistes simultàniament.
- Reproducció independent d'un sol tall.
- Regulació manual de l'envolupant dels talls d'àudio.
- Regulació automàtica de l'envolupant segons un sistema de prioritització de les pistes.
- 9 modes diferents de reproducció a partir d'un tram seleccionat del projecte.
- Desat dels projectes en format XML per a la seva posterior recuperació, o bé com a arxiu d'àudio únic.
- Funcions de desfer i refer.
- Zoom sobre les pistes.

La programació s'ha dividit en dues parts diferenciades, el nucli i la interfície. El nucli es pot considerar com un programa independent sense cap part visible, mentre que la interfície és la que permet a l'usuari utilitzar la majoria de funcions del nucli amb una representació gràfica dels seus elements.

Tot el conjunt ha estat programat amb Visual Basic 6.0 i, concretament, els elements del nucli s'han creat utilitzant classes i col·leccions, mentre que a la interfície s'han utilitzat controls d'usuari.

Title: Creation of a multitrack system with 4 sound tracks for the DSI Sound

Authors: Oriol Príncipe Dols
Álvaro del Río Navarro

Director: Xavier Montero Amorós

Date: September 5th 2005

Overview

The aim of this program is the creation of reports and chronicles as audio projects, by using several sound tracks, for their later radio broadcasting. The application is focalized to journalists who work in a radio station office and don't necessarily have a high knowledge of computers. The program contains a friendly interface that lets the user work with several tracks where previously recorded audio files can be added. Its main features are:

- Possibility to add, move, define an interval or remove audio cuts from a track.
- Drag & Drop method to add audio cuts from any location.
- Playback of one or several tracks simultaneously.
- Playback of one single cut.
- Manual control of the audio cuts envelope.
- Automatic control of the envelope according to a track priority system.
- 9 different playback modes depending on the project selected range.
- Projects can be saved in XML format for its later edition, or in a single audio file.
- Undo and redo functionalities.
- Zoom into the tracks.

The programming was divided into two different parts, the kernel and the interface. The kernel can be considered as an independent program with no visible part, while the interface is what lets the user use the majority of the kernel functions thanks to a graphical representation of its elements.

The whole project was programmed with Visual Basic 6.0 and, to be specific, we used classes and collections for the kernel elements, whereas we used user controls to build the interface.

ÍNDIX

INTRODUCCIÓ	1
CAPÍTOL 1. REQUERIMENTS.....	2
1.1. Documentació de referència.....	2
1.2. Definicions de termes segons la interpretació que tindran en aquest document.....	2
1.3. Objectius generals	4
1.4. Requeriments del DSI Visual Track – Kernel	4
1.5. Requeriments del DSI Visual Track - Interfície.....	6
1.6. Requeriments desitjables	9
CAPÍTOL 2. FUNCIONALITATS I CARACTERÍSTIQUES.....	12
2.1. Seccions de la pantalla principal del DSI Visual Track.....	12
2.1.1. El menú superior.....	12
2.1.2. El navegador d'arxius.....	15
2.1.3. El menú inferior.....	15
2.1.4. Les pistes	17
2.1.5. Els talls	19
2.1.6. Els punts de control	21
CAPÍTOL 3. ARQUITECTURA	22
3.1. El nucli.....	22
3.2. La interfície.....	22
3.2.1. Els components	23
3.2.2. L'executable	23
CAPÍTOL 4. FUNCIONAMENT INTERN DEL PROGRAMA	24
4.1. Kernel.....	24
4.1.1. Introducció.....	24
4.1.2. DirectShow Editing Services	27
4.1.3. Aspectes i funcions destacables del kernel.....	30
4.2. Interfície.....	35
4.2.1. Introducció.....	35
4.2.2. Aspectes i funcions destacables de la interfície.....	38
CONCLUSIONS I PROPOSTES FUTURES	41
BIBLIOGRAFIA.....	43

ANNEXOS	44
1. Descripció de les classes del nucli	44
2. Descripció dels controls de la interfície	61
3. Objecte AMTimelineObj	80

INTRODUCCIÓ

El DSI Visual Track neix a partir d'una necessitat de Ràdio Sant Cugat per formar part del DSI Sound Station, una suite de programes dedicats a l'àudio enfocada a l'emissió radiofònica. El nostre programa en particular pretén cobrir la part dels muntatges d'àudio que es podrien fer en una redacció.

Actualment al mercat existeixen diversos programes que compleixen els requeriments necessaris per realitzar aquest tipus de funcions, però que per una ràdio sense gaires recursos econòmics es fan inaccessibles.

Amb la voluntat de facilitar aquesta tecnologia a totes les ràdios petites o mitjanes empreses que requereixin l'ús d'un multipistes hem ideat el DSI Visual Track, un programa de lliure distribució basat en la filosofia GNU (GPL), que permet a qualsevol usuari la utilització gratuïta del programa i a qualsevol programador utilitzar, també gratuïtament, parcialment o íntegrament el codi font del programa.

L'aplicació ha estat generada tenint en compte la possibilitat de que l'usuari, en principi periodista de professió, no tingui profunds coneixements informàtics. Basant-nos en aquesta premissa hem dissenyat una aplicació amb un interfície senzilla que permetrà fer-se amb les funcionalitats bàsiques del programa de forma intuïtiva. A través de la nostra pròpia experiència en la utilització de programes d'àudio com l'Adobe® Audition™ (antigament Cool Edit®) i les mancances trobades en programes comercials com ara el Dalet, hem fet una síntesi de les característiques més importants i les hem volgut simplificar per facilitar-ne l'ús.

Nota: Adobe® és marca registrada propietat d'Adobe.

Cool Edit® és marca registrada propietat de Syntrillium.

Dalet és marca registrada propietat de Dalet Digital Media Systems.

CAPÍTOL 1. REQUERIMENTS

Una de les parts més importants en començar un projecte de programació és definir exactament amb quin objectiu es crea, quines necessitats ha de cobrir i a través de quins mitjans es realitzaran.

El contingut d'aquest capítol és, per tant, el document amb els requeriments que es van plantejar a l'inici del projecte, encara que amb certes modificacions. Durant la realització del projecte, alguns dels requeriments han passat a ser desitjables en comptes de necessaris, altres s'han ampliat i fins i tot se n'han hagut d'afegir per poder cobrir funcionalitats bàsiques que al principi no s'havien tingut en compte.

A la data d'entrega del projecte, considerem que el programa es troba a la versió 0.9. Els requeriments que no han estat assolits es deuen a certes indefinicions per part de Ràdio Sant Cugat en algunes funcionalitats del programa i a la falta de temps degut a les dimensions que aquest ha anat adquirint.

Un cop l'apartat de requeriments desitjables es trobés implementat, podríem considerar que el programa es trobaria a la versió 1.0.

Per tenir un coneixement més exhaustiu de les característiques i capacitats del programa recomanem al lector que llegeixi com a complement dels requeriments el capítol 2 "Funcionalitats i característiques".

1.1. Documentació de referència

El marc on s'engloba aquest projecte és el conjunt d'aplicacions relacionades amb un software per a emissores de ràdio anomenat DSI Sound Station. DSI Sound Station neix com un software de gestió de l'àudio en una emissora de ràdio i al llarg dels anys s'extén i amplia mitjançant la creació de múltiples aplicacions addicionals, entre les quals trobem l'objecte del present projecte.

Per entendre el marc de referència, caldrà al lector visitar la web <http://www.sound-station.info/> En el present document s'assumeix llegida i entesa la part d'informació referent a DSI Sound Station Kernel i DSI Sound Station Interface que es troben a la web.

1.2. Definicions de termes segons la interpretació que tindran en aquest document

MSDN: Llibreria d'informació tècnica propietat de Microsoft Corporation accessible públicament via web a <http://msdn.microsoft.com/library/>

Arxiu de so: Arxiu estàndard de tipus RIFF/WAV, habitualment emmagatzemat sobre disc dur, que conté dades de so en format lineal o comprimit en MPEG-2. Per a més informació sobre aquest tipus d'arxiu, vegeu MSDN.

Tall/Corte/Cut: Tram d'un arxiu d'àudio susceptible de ser reproduït.

Pista/Pista/Track: Espai on s'insereixen els talls en una posició determinada, per a seqüenciar i controlar la seva reproducció.

Projecte/Proyecto/Project (accepció 1): Conjunt de diverses pistes per a la reproducció simultània i sincronitzada de totes elles. El projecte conserva les característiques de las pistes per a una utilització posterior.

Projecte/Proyecto/Project (accepció 2): El desenvolupament del software en si mateix. No s'ha de confondre projecte en el sentit de "conjunt de sons" (accepció 1) amb projecte com a desenvolupament del software (accepció 2).

Navegador de cartutxos/Cart Browser: Espai des d'on es visualitzen els talls de so nadius del sistema DSI Sound Station per poder-los inserir a les pistes.

Drag & Drop: Sistema de selecció i arrossegament d'elements propi del sistema operatiu Windows que permet seleccionar un objecte i arrossegar-lo al damunt d'un altre per fer una acció conjunta entre objecte arrossegat i destí del moviment. Consisteix en clicar amb el botó principal a l'element a arrossegar i, mantenint el botó premut, desplaçar l'element fins al lloc de destí i acte seguit deixar anar el botó del ratolí.

API: Application Programming Interface. Llista documentada de crides a funcions que un programa (anomenat servidor o llibreria) ofereix als programadors perquè construeixin les aplicacions que utilitzen els algorismes i la lògica creats en el primer (servidor o llibreria).

Servidor o Llibreria: Aplicació que ofereix una API.

Client: Aplicació que utilitza els recursos d'un servidor fent us de la seva API per enviar dades a processar i recuperar el resultat del procés.

Volum instantani: Volum que va prenent, en funció del temps, cada un dels instants possibles d'una pista que representa so.

Renderitzar: Interpretar les dades d'un projecte de so multipista, per tal de calcular i salvar a disc un únic arxiu de so que sona igual que la reproducció del projecte complet.

Barra d'eines: Conjunt de diverses icones en fila, una al costat de l'altra, que permeten als usuaris d'un programa utilitzar el ratolí per fer directament diverses accions associades, una per botó, o més d'una si el botó desplega un menú d'opcions.

1.3. Objectius generals

RK-G-1 La finalitat d'aquest projecte és dotar a l'emissora de ràdio amb un programa que permeti la creació de muntatges sonors mitjançant la creació i edició de projectes d'àudio, utilitzant diverses pistes de so, i inserint talls a cada pista. Els talls poden provenir del DSI Sound Station o d'arxius externs. El muntatge resultant es podrà desar com a un so únic per a la seva posterior emissió radiofònica.

NOTA: Donat el RK-G-1, en tot aquest document s'utilitzaran com a perfectament intercanviables els termes de "projecte" (entès no com un projecte de so sinó com el projecte de desenvolupament del software), "programa" (entès com a eina informàtica que compleix l'anomenat objectiu), "DSI Visual Track" (nom assignat al projecte) i "Editor multipista" (denominació genèrica que descriuria el tipus de software que es pretén desenvolupar).

RK-G-2 El programa funcionarà sota el sistema operatiu Windows 2000 i Windows XP.

RK-G-3 El DSI Visual Track constarà de dues parts diferenciades.

- **DSI Visual Track Kernel:** Serà el nucli de funcionament de l'editor multipista, i tindrà la responsabilitat d'integrar totes les funcions relatives a l'edició (com l'ordenació de talls en una pista o la reproducció del projecte). Funcionarà com una aplicació servidora perquè una aplicació client utilitzi l'API del Kernel per cridar les seves funcions. Aquest tros del programa no té cap element gràfic. Conté únicament algorismes de funcionament i lògica.
- **DSI Visual Track Interface:** Serà el programa client que consumirà els recursos del Kernel. La seva missió es construir els gràfics en pantalla per representar les dades del Kernel, que s'obtidran mitjançant crides a l'API del mateix. Els dispositius d'entrada (teclat i ratolí) també s'utilitzaran per donar ordres al Kernel mitjançant crides a l'API.

RK-G-4 El DSI Visual Track arrencarà a través d'un executable (.exe) que correspondrà, en particular, a l'executable de la interfície.

1.4. Requeriments del DSI Visual Track – Kernel

RK-K-1 El projecte s'inicialitza sense cap pista, a l'espera de les que necessiti la interfície.

RK-K-2 Permetrà la reproducció (Play) de talls d'àudio digital i la seva interrupció amb la funció Stop.

RK-K-3 Es podran afegir i suprimir talls a qualsevol pista.

RK-K-4 Es podran desplaçar talls dins de la mateixa pista sempre que no se superposin entre si.

RK-K-5 El nucli suportarà l'ús d'un màxim de 32.767 pistes.

RK-K-6 Es podran definir diferents punts de control dintre de cada tall, que podran ser utilitzats, entre d'altres funcions, per definir l'envolupant del seu volum.

RK-K-7 Es podrà fixar si aquests punts de control seran editables o no.

RK-K-8 Els punts podran ser eliminats quan es permeti la seva edició.

RK-K-9 El rang de volum anirà des del valor nominal de cada tall fins a -40 dB (llindar que considerem com a silenci).

RK-K-10 Es podrà seleccionar l'interval que serà reproduït de cada tall, sense que això afecti l'arxiu original de so.

RK-K-11 Es podrà escollir quines pistes se sentiran i quines no. Cada pista podrà estar en un dels següents estats:

- Normal: La pista se sent, sempre i quan no hi hagi una altra pista amb la propietat "solo" activada.
- Solo: Se sent només aquesta pista, silenciant les altres que estiguin en estat "Normal".
- Mute: La pista està silenciada.

RK-K-12 Es podran fer reproduccions totals del projecte per una sortida d'àudio del PC on s'executi.

RK-K-13 Es podran fer reproduccions parcials del projecte, especificant-ne un interval, per una sortida d'àudio del PC on s'executi.

RK-K-14 Qualsevol tall serà susceptible de ser eliminat.

RK-K-15 El temps de reproducció en pista serà de 2.147.483.647 milisegons (596,52 hores).

RK-K-16 Es podran desfer i refer els canvis que s'hagin anat aplicant al projecte.

RK-K-17 Es podrà salvar l'estat del projecte per continuar-hi treballant posteriorment havent tancat el programa. Les dades es salvaran en format XML.

RK-K-18 Es podrà obrir un projecte prèviament guardat en format XML.

RK-K-19 En qualsevol moment es podrà crear un nou projecte buit i tancar l'anterior.

RK-K-20 El programa client podrà consultar al Kernel si el projecte carregat ha estat editat o no, amb l'objectiu de poder-ne visualitzar l'estat i, en cas necessari, formular una pregunta de confirmació a l'usuari abans de tancar un projecte editat sobre el que es perdrien els canvis.

RK-K-21 El Kernel serà capaç de reproduir, com a mínim, arxius gravats en format MP2 a 256Kbps, Stereo o Joint Stereo.

RK-K-22 El Kernel presentarà una API mitjançant la tecnologia ActiveX, s'executarà en la seva modalitat "in process" (DLL Active X) i serà cridable des d'un client escrit en Visual Basic.

RK-K-23 El Kernel, en la mesura del possible, estarà escrit en llenguatge Visual Basic.

1.5. Requeriments del DSI Visual Track - Interfície

RK-I-1 La interfície s'encarregarà de la comunicació entre l'usuari i el nucli. Aquesta es relacionarà amb el nucli mitjançant crides a les seves funcions específiques.

RK-I-2 La interfície estarà dissenyat de tal forma que pugui funcionar en mode "standalone" (executable normal o Standard Exe) o bé integrat en una altra aplicació en forma de control d'usuari (Control ActiveX). En particular, en la mesura del possible la lògica dels requeriments es col·locarà al Control ActiveX de forma que sigui configurable què es veu i què no i aquest control pugui ser integrat en una aplicació de tercers o en un executable que es limitarà a instanciar el control.

RK-I-3 En executar el programa, el DSI Visual Track mostrarà una pantalla de presentació mentre es carrega el programa.

RK-I-4 La col·locació de sons a les pistes es realitzarà mitjançant el mètode Drag&Drop arrossegant un arxiu del sistema operatiu sobre una pista.

RK-I-5 La interfície proveirà a l'usuari amb un navegador d'arxius per tal de facilitar l'inserció de sons en cas de que no vulgui utilitzar l'explorador del sistema operatiu.

RK-I-6 En cas que el control d'usuari sigui instanciat per una aplicació de tercers, aquesta aplicació externa podrà controlar la interfície de forma que sigui impossible mostrar el navegador d'arxius.

RK-I-7 En iniciar-se la interfície, la pantalla principal del DSI Visual Track es visualitzarà a pantalla completa.

RK-I-8 En iniciar-se la interfície, si aquesta s'inicia en mode "stand-alone", aquesta crearà un nou projecte amb un nombre prefixat de pistes. En cas d'iniciar-se com a control d'una tercera aplicació, serà aquesta última la que controlarà el nombre inicial de pistes.

RK-I-9 Una vegada generat el projecte, la interfície del programa tindrà el següent aspecte:

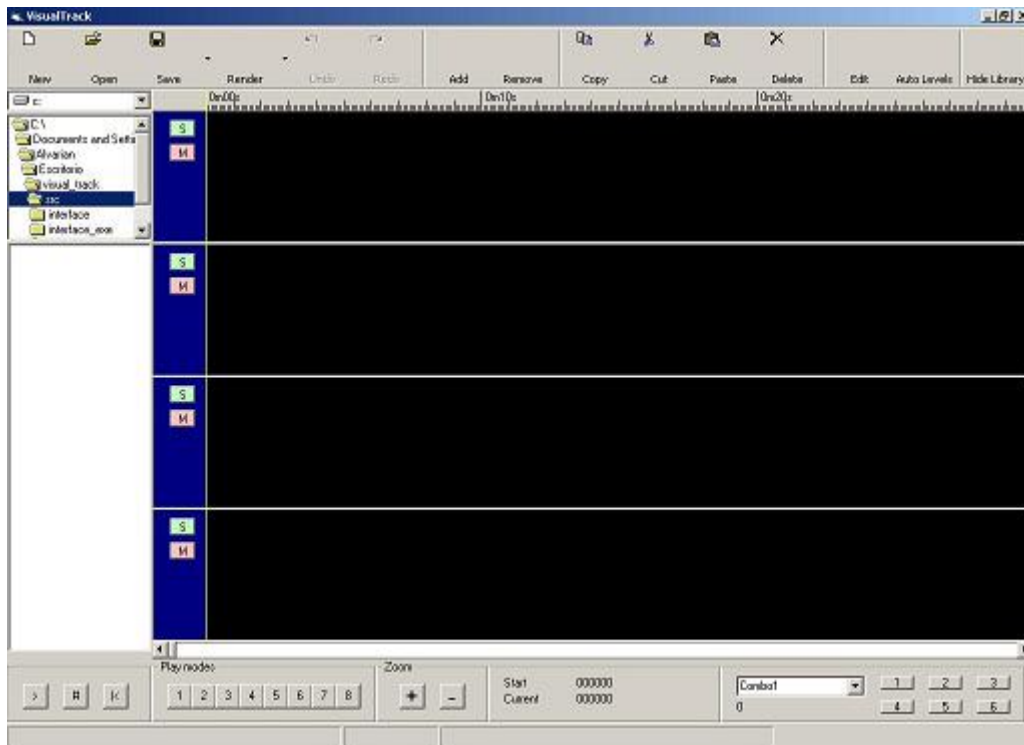


Fig 1.1 Aspecte general de la interfície

RK-I-10 L'àrea de les pistes serà redimensionable tant en llargada com en alçada.

RK-I-11 Posteriorment es podran afegir o eliminar pistes, amb un mínim d'una i un màxim de sis pistes.

RK-I-12 Els talls d'àudio quedaran representats en forma de bloc dins de la pista.

RK-I-13 En cas que l'arxiu original a visualitzar en un bloc estigui en format MP2 amb energia enregistrada a l'interior de cada trama, llavors, dins de cada bloc s'hi representarà la forma d'ona aproximada de l'arxiu d'àudio corresponent, col·locant una barra vertical per bloc d'energia comprimida.

RK-I-14 Els blocs amb els talls de so, s'inseriran a partir del punt on se situï el ratolí en fer el Drag&Drop.

RK-I-15 Cada pista permetrà la inserció de múltiples talls, tot i que no es podran superposar.

RK-I-16 Els talls es podran desplaçar manualment dins de la pròpia pista, sempre que no se superposin, utilitzant el botó principal del ratolí.

RK-I-17 Mentre un tall es desplaça per la pista, la representació en forma d'ona es continuarà veient dins del seu bloc.

RK-I-18 El nivell de volum de cada tall d'àudio estarà representat per una línia que mostrarà l'evolució de l'envolupant de l'ona representada, és a dir, el nivell de volum instantani. Aquesta línia estarà formada a partir de la unió dels punts de volum afegits.

RK-I-19 L'envolupant es modificarà utilitzant el botó principal del ratolí, clicant i arrossegant des de qualsevol punt de volum de l'envolupant.

RK-I-20 Inicialment la línia de l'envolupant estarà definida per dos punts a l'inici i al final amb el nivell a 0 dB. Posteriorment se'n podran definir més.

RK-I-21 A la part superior de la interfície hi haurà una barra d'eines.

RK-I-22 De cada tall d'àudio se'n podrà seleccionar quin interval es reproduirà al projecte.

RK-I-23 Es podrà indicar el punt d'inici de reproducció del projecte mitjançant un marcador.

RK-I-24 Una vegada iniciada la reproducció, una barra de desplaçament anirà indicant el curs de la reproducció.

RK-I-25 Es podrà seleccionar una secció del projecte per tal de reproduir-la sense que això afecti el seu estat.

RK-I-26 Es podran suprimir els talls d'àudio amb l'ús del botó Delete.

RK-I-27 Cada pista disposarà dels botons "Solo" i "Mute", que representaran l'estat de la pista al Kernel i permetran modificar-lo.

RK-I-28 Per a desactivar el botó Solo d'una pista caldrà prémer aquest mateix botó o activar el Solo d'una altra pista.

RK-I-29 Per a ometre la reproducció d'una pista o més pistes en concret, s'utilitzarà el botó Mute de cadascuna d'elles.

RK-I-30 Per a desactivar botons Mute caldrà tornar-los a prémer o bé marcar el botó Solo d'alguna.

RK-I-31 Es podran desfer i refer els canvis que s'hagin anat aplicant al projecte utilitzant un parell de botons (Undo i Redo).

RK-I-32 Existirà un botó de Save Project per a guardar l'estat general del projecte.

RK-I-33 Existirà un botó de Render Project per a guardar el projecte com un sol arxiu d'àudio.

RK-I-34 En la mesura del possible, la interfície estarà escrit en llenguatge Visual Basic.

1.6. Requeriments desitjables

(Per a la versió 1.0 del DSI Visual Track)

RK-D-1 L'usuari podrà triar entre dos modes de control per als nivells de volum instantanis dins de cada pista:

- Mode manual (Automatic Levels = Off): L'usuari controla el nivell de so de cada tall amb el ratolí, establint punts de control entre els quals s'interpola linealment el so.
- Mode automàtic (Automatic Levels = On): L'usuari no es preocupa dels nivells de so. Cada pista té una certa prioritats. Quan dues o més pistes es solapen, les de més prioritats sonen a nivell normal (0dB) mentre que les menys prioritats sonen a volums més baixos automàticament. En el moment que una pista d'alta prioritats es queda sense so, si una pista de prioritats inferior encara conté material audible, puja automàticament el volum.

Quan l'opció Automatic Levels estigui activada:

RK-D-2 Cada pista tindrà un nivell de prioritats, representat per un nombre natural, incloent el zero.

RK-D-3 L'usuari podrà assignar les prioritats a les pistes.

RK-D-4 Els nivells de so corresponents a cada prioritats (on 0 és la més alta i 3 la més baixa) seran els següents: Prioritats 0 = 0dB; Prioritats 1 = -6dB; Prioritats 2 = -10dB; Prioritats 3 = -20dB.

RK-D-5 Aquests nivells de so es veuran representats amb l'envolupant i no es podran modificar manualment.

RK-D-6 La prioritats més baixa que se li pot assignar a un tram d'una pista és la 3.

RK-D-7 L'opció d'automatic levels es podrà activar o desactivar mitjançant un botó de la barra d'eines.

Si l'opció Automatic Levels es troba desactivada:

RK-D-8 Totes les pistes es reproduiran sense tenir en compte les prioritats (0 dB per defecte).

RK-D-9 En aquest cas, serà possible fer variar manualment els volums de reproducció de cadascuna de les pistes en diferents intervals de temps modificant-ne l'envolupant.

RK-D-10 Per poder modificar l'envolupant, serà necessari activar el botó Edit Envelope, que només serà actiu quan l'opció Automatic Levels estigui desactivada.

RK-D-11 El volum de l'àudio entre dos punts de control consecutius serà una interpolació d'aquests dos.

RK-D-12 Es podran copiar talls de so prèviament afegits a una pista per després enganxar-los en qualsevol altra.

RK-D-13 Es podran copiar i tallar trams d'àudio seleccionats per després enganxar-los en qualsevol pista.

RK-D-14 Es podrà renderitzar el projecte com un sol arxiu d'àudio.

RK-D-15 El renderitzat generarà arxius en format MP2 (MPEG I – Layer II) a 256Kbps, Stereo o Joint Stereo, amb nivell d'energia emmagatzemat dins de les trames MPEG.

RK-D-16 El DSI Visual Track integrarà un display on es visualitzarà el temps de la reproducció en curs i els temps d'inici i final de possibles seleccions.

RK-D-17 Magnetic Positions:

- Al moure un tall, s'enganxarà a posicions magnètiques per evitar que es quedi separat del tall del costat.
- Les posicions magnètiques són: "Posició 0", "Just abans del primer tall", "Just després del primer tall", "Just abans del segon tall", "Just després del segon tall", "Just abans del tercer tall", etc.
- Per cada posició es calcularà si és possible moure'l a la posició magnètica. Si és possible, quan el ratolí posicioni el tall a menys d'un nombre de píxels (per exemple 10 píxels) de distància d'una posició magnètica, el tall el col·locarà a aquesta posició.

- Aquesta opció s'activarà o desactivarà amb un botó de la barra d'eines (per exemple "Magnetic Positions").

CAPÍTOL 2. FUNCIONALITATS I CARACTERÍSTIQUES

En els següents paràgrafs detallarem totes les funcions de les quals disposa actualment el DSI Visual Track, per tal que qualsevol persona que hi pugui estar interessada es faci una idea de quines opcions li pot oferir aquest editor multipista.

2.1. Seccions de la pantalla principal del DSI Visual Track

A la pantalla principal del programa hi podem diferenciar quatre zones:

- 1- El menú superior
- 2- El navegador d'arxius
- 3- El menú inferior
- 4- Les pistes

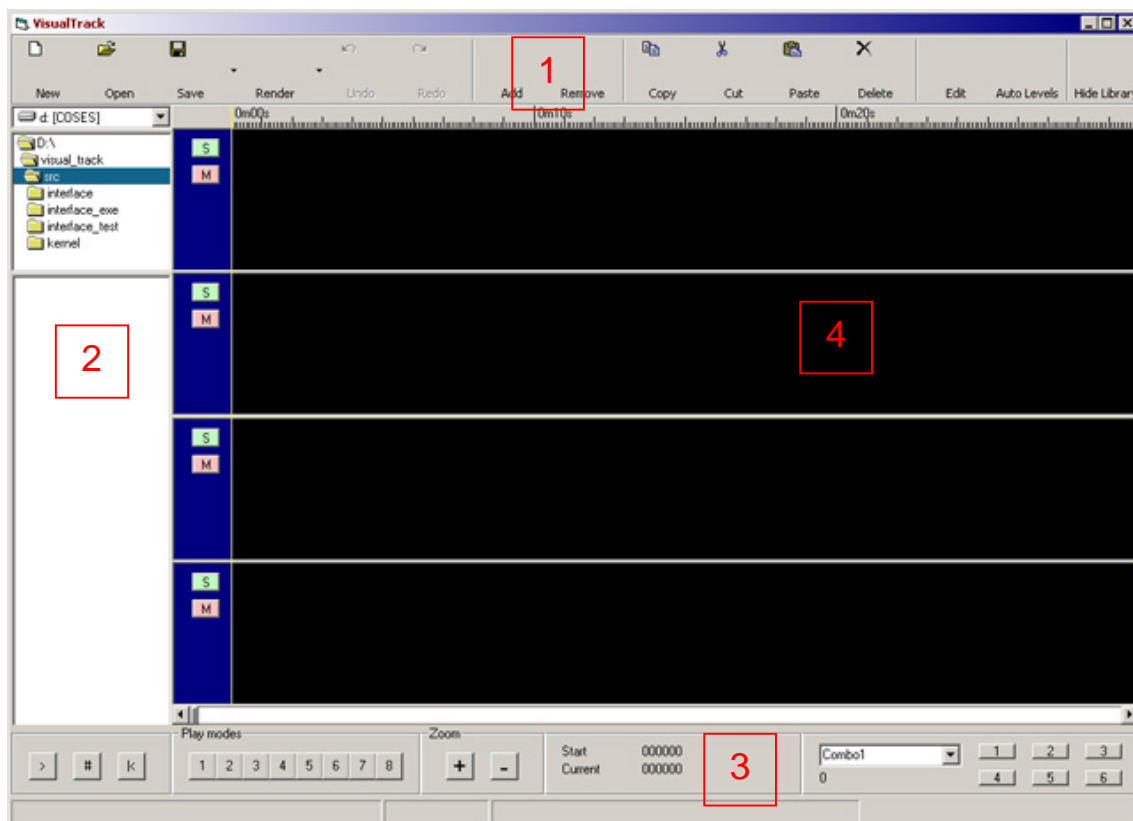


Fig 2.1 Pantalla principal del DSI Visual Track

2.1.1. El menú superior

Com es pot veure a la Fig 2.2 el menú superior es divideix en 5 parts diferenciades.

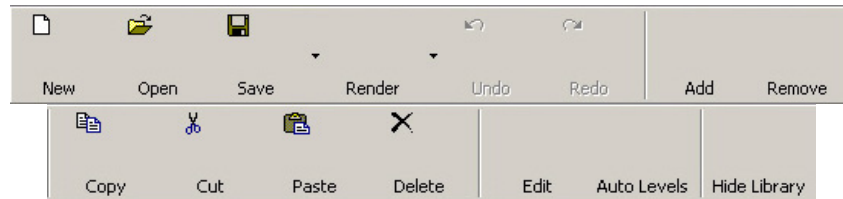


Fig 2.2 Menú superior del DSI Visual Track

El primer bloc consta dels botons relatius al projecte en general. Concretament són:

New: crea un nou projecte buit, i en el cas que el projecte anterior no s'hagués guardat apareix una finestra emergent preguntant si es desitja fer-ho. El nou projecte es generarà amb tantes pistes lliures com tingui definides la interfície.

Open: obre un projecte prèviament desat pel DSI Visual Track. Una vegada premut ens trobarem amb la següent pantalla, que només ens permetrà obrir arxius que siguin compatibles amb el format de projecte:

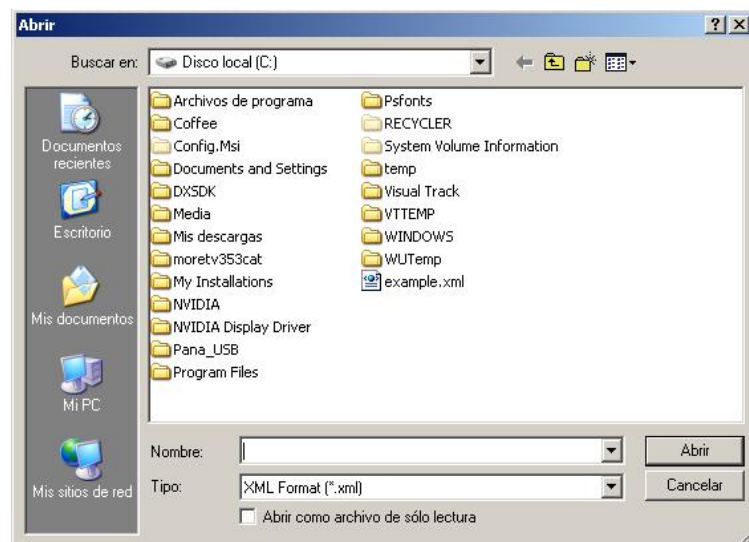


Fig 2.3 Pantalla d'*Open* estàndard del Windows

Save: desa el projecte amb el que s'està treballant per tal de poder-lo recuperar en qualsevol altre moment. Si és la primera vegada que es vol salvar el projecte actual, veurem directament la següent pantalla amb l'objectiu de donar-li una carpeta de destí i un nom a l'arxiu.

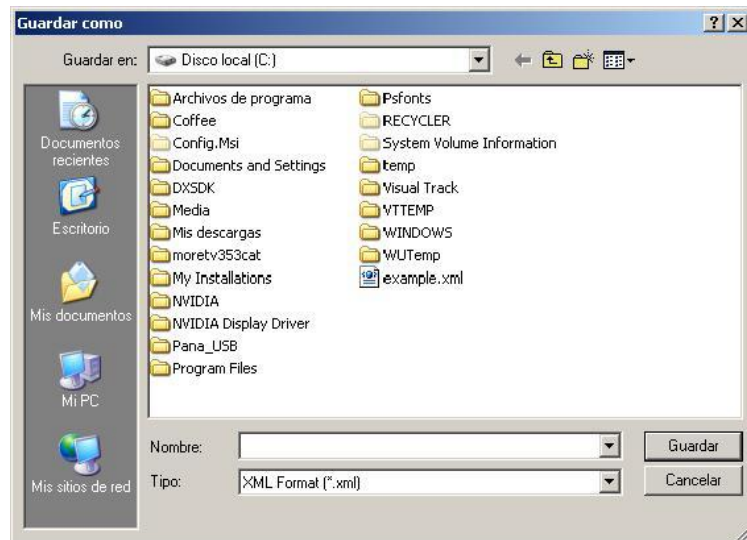


Fig 2.4 Pantalla de Save estàndard del Windows

Quan prèviament ja se li hagi donat un nom al projecte, el programa el guardarà automàticament al mateix arxiu sobreescrivint la nova informació. En el cas que es vulgui guardar el projecte amb un altre nom d'arxiu només caldrà seleccionar l'opció "Save as" utilitzant el menú desplegable que incorpora el botó "Save".

Render: crea un únic arxiu de so de format MPEG-2 com a resultat de la composició sencera a partir dels diversos talls d'àudio. Cal tenir en compte que a partir d'aquest moment aquest arxiu només es podrà utilitzar com a únic tall de so, ja que no s'emmagatzema la informació de les pistes per separat (no implementat).

Undo: desfà l'última acció que haguem efectuat sobre les pistes o els talls de so.

Redo: refà l'última acció que haguem desfet utilitzant el botó Undo.

El segon bloc de botons fa referència a les pistes. Els botons associats són:

Add: afegeix una pista buida a sota de l'última que es vegi en pantalla.

Remove: elimina l'última pista junt amb tots els talls que contingui.

El tercer bloc fa referència als talls i les accions que es poden portar a terme:

Copy, Cut i Paste: en un principi, Ràdio Sant Cugat va proposar que el programa contingués les funcions de copiar, tallar i enganxar. Tot i això, aquests mètodes no estan implementats a causa de la indefinició per part de la ràdio de la funció exacta que aquests haurien de fer, ja que es podien entendre de diverses maneres: copiar un tram de totes les pistes, copiar un tram d'una sola pista, copiar talls sencers o copiar un tram d'un tall.

Delete: elimina el tall de la pista activa que estigui seleccionat. La tecla suprimir també realitza aquesta acció.

El quart apartat del menú superior està relacionat amb els punts de control:

Edit: activa o desactiva l'opció d'editar l'envolupant dels talls de so mitjançant els punts de control de volum.

Auto Levels: activa o desactiva l'opció *Automatic Levels*, que permet ajustar automàticament el volum de cada tall de so per adaptar-lo a la resta del projecte (no implementat).

L'últim botó del menú fa referència a la llibreria d'arxius:

Hide Library: Amaga o mostra el navegador d'arxius.

2.1.2. El navegador d'arxius

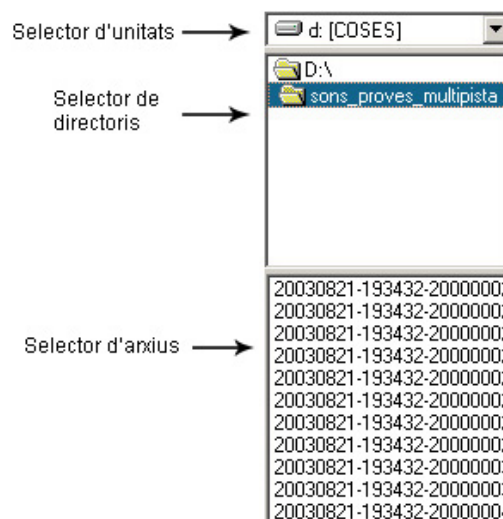


Fig 2.5 Navegador d'arxius del DSI Visual Track

El navegador d'arxius és l'eina que ens permet afegir talls de so a les pistes. Un cop hem localitzat un arxiu amb el que volem treballar només cal clicar amb el botó esquerre sobre el nom de l'arxiu i arrossegat-lo fins a la pista per deixar-lo anar a la posició de l'eix temporal on el vulguem col·locar.

2.1.3. El menú inferior

Aquest menú és el que ens permet fer els diferents tipus de reproduccions del projecte, fer zooms sobre una zona concreta i posicionar *markers* en forma de barres per tenir controlats diferents punts temporals del projecte.



Fig 2.6 Menú inferior del DSI Visual Track

Les diferents opcions que s'ofereixen són:

Opcions de reproducció:

Play: reproduïx el projecte des de la posició marcada per la barra d'inici i fins al final del projecte. Quan hi ha un tram seleccionat, només es farà la reproducció de la selecció. L'acció de "Play" també es pot realitzar utilitzant la barra espaciadora.

Stop: atura qualsevol reproducció que s'estigui efectuant.

Go to beginning: Situa la barra d'inici al principi del projecte.

Modes de reproducció:

Play mode 1: reproduïx tot el projecte sencer.

Play mode 2: reproduïx la zona visible en pantalla del projecte.

Play mode 3: reproduïx el projecte des del començament i fins a l'inici de la selecció.

Play mode 4: reproduïx el projecte des de l'inici de la selecció.

Play mode 5: reproduïx el projecte des del començament i fins al final de la selecció.

Play mode 6: reproduïx el projecte des del final de la selecció.

Play mode 7: reproduïx tot el projecte excepte la zona seleccionada.

Play mode 8: reproduïx uns segons abans i després de la zona seleccionada.

Zoom:

Zoom +: amplia la visió d'un tram de les pistes que hi hagi en pantalla. El centre del zoom s'aplica sobre la posició on es trobi la barra d'inici de reproducció.

Zoom -: augmenta l'interval de temps visible en pantalla. Com en el "Zoom +" l'efecte del zoom es fa centrat sobre la barra d'inici de reproducció.

Displays de temps:

Start: mostra la posició de la barra d'inici en milisegons.

Current: mostra el temps actual de reproducció expressat en milisegons.

Markers: Aquest apartat tampoc està implementat perquè va ser un requeriment desitjable que es va afegir a l'etapa final del projecte.

2.1.4. Les pistes

Les pistes són l'espai on s'insereixen els talls de so. Representen un espai temporal de proporcions relatives al nivell de zoom que se'ls apliqui.



Fig 2.7 Pistes del DSI Visual Track

Per ajudar-nos a treballar amb les pistes, tenim una sèrie d'elements (veure Fig 2.7):

- A la part superior, una barra de temps ens serveix per conèixer la durada temporal d'allò que estem visualitzant.
- A la part inferior, trobem una barra de desplaçament que ens permet moure'ns per tota la durada del projecte.

- A l'esquerra de cada pista, un boto de *Solo* per poder escoltar una sola pista i silenciar la resta i un de *Mute* per poder silenciar les que vulguem. El botó de Solo impera sobre els de Mute, així si ens trobem que premem el Solo en una pista que estava en Mute aquesta canviarà a Solo i serà l'única que se sentirà.
- Un altre dels elements és la barra d'inici amb la qual podem marcar, utilitzant el botó esquerra del ratolí, el punt en què volem que comenci la reproducció del projecte.
- L'últim element és la barra de reproducció, només visible quan hi ha una reproducció en curs, donant-nos la posició instantània de temps de la reproducció.

Accions que es poden efectuar sobre les pistes:

- *Inserció de talls*: Tal com està explicat a l'apartat del navegador d'arxius, una de les principals accions és la d'afegir talls utilitzant el mètode de Drag & Drop amb els arxius mostrats pel navegador.
- *Selecció*: Si volem marcar un punt d'inici i un de final, amb el ratolí podem fer una selecció d'un tram del projecte que serà la que sentirem en reproduir-lo amb el play general. Per fer l'acció de selecció només cal clicar en una part buida de les pistes (o sobre un tall si no està l'Edit Envelope activat) i arrossegar fins la posició desitjada. Les pistes i els talls quedaran ressaltats de la següent forma per indicar quin és el tram seleccionat:

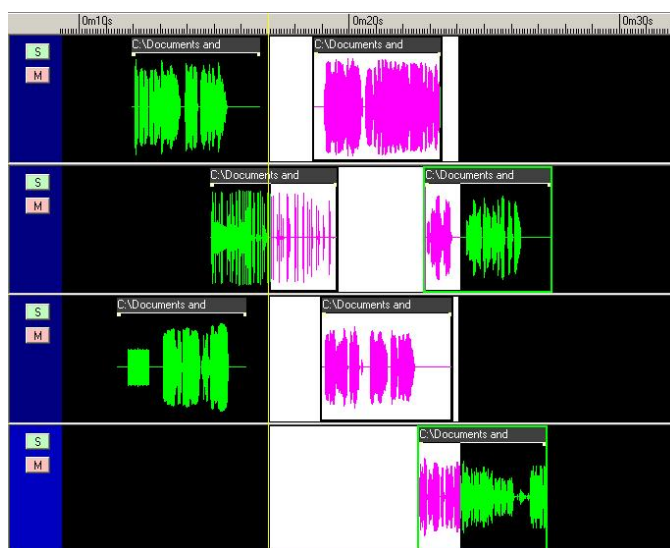


Fig 2.8 Pistes amb una zona seleccionada

- *Pista activa*: Quan es cliqui sobre una pista o un dels seus talls, la pista en qüestió passarà a ser la pista activa, cosa que ens permetrà efectuar accions exclusivament en aquella pista, com l'eliminació d'un tall. Sabrem quina és la pista activa quan la zona blava que alberga els botons de Mute i Solo estigui ressaltada, com s'aprecia en la primera pista de la següent imatge:

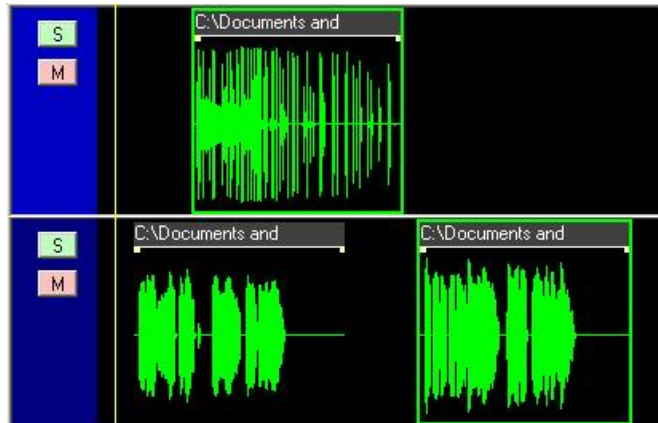


Fig 2.9 Diferenciació entre la pista activa i una de no activa

2.1.5. Els talls

Quan inserim un tall en una pista des del navegador d'arxius, veiem el següent:

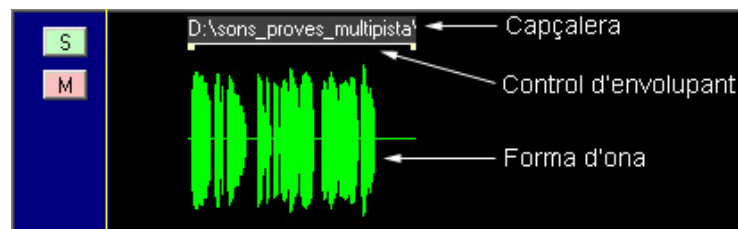


Fig 2.10 Pista amb un tall

Cada tall està format per les tres parts assenyalades a la figura 3.10:

- *La forma d'ona*: constitueix la major part del tall i representa una forma aproximada del seu so. Com més gran sigui l'amplitud del dibuix, més fort se sentirà, i com més proper a la línia que divideix l'ona, més fluix se sentirà.
- *Capçalera*: aquesta zona ens mostra la localització i el nom de l'arxiu.

- *Control d'envolupant*: aquesta part està formada pels punts de control i una línia que els uneix. Aquesta línia és la que representa l'envolupant de volum del tall.

Un cop hem afegit el tall, podem fer diferents accions sobre ell:

- *Seleccionar-lo*: si amb el botó esquerre del ratolí cliquem una sola vegada damunt de la capçalera, veurem que el tall queda seleccionat amb un requadre de color verd al seu voltant.
- *Desplaçar-lo*: amb el ratolí el podem moure a esquerra i dreta al llarg de la pista en la que es troba el tall. Per fer-ho, hem de situar el ratolí damunt de la capçalera (veurem que la icona del ratolí es converteix en una creu), clicar amb el botó esquerre i desplaçar el tall mentre mantenim el botó clicat. L'única restricció és que dos talls no es poden superposar en la mateixa pista.
- *Menú contextual del tall*: si cliquem a la capçalera amb el botó dret, despleguem una menú que ens permetrà reproduir l'interval visible d'aquell tall, aturar aquesta reproducció o amagar aquest menú.
- *Seleccionar-ne un interval*: d'aquesta manera podem escollir quina part del tall volem utilitzar en el nostre projecte. El podem reduir tan per la dreta com per l'esquerra. Per fer-ho, cal situar el ratolí al marge dret o esquerre del tall (veurem que la icona del ratolí es converteix en una fletxa doble), clicar amb el botó esquerre i seleccionar l'interval que desitgem mentre mantenim el botó clicat.

En la següent imatge podem veure el tall de la figura 3.6 mentre s'està efectuant la selecció d'un interval:



Fig 2.11 Tall amb interval seleccionat

- *Eliminar-lo de la pista*: utilitzant el botó corresponent del menú superior o bé prement la tecla suprimir, podem eliminar el tall que estigui seleccionat de la pista activa.

2.1.6. Els punts de control

Dins de cada tall hi trobem els punts de control, que serveixen per representar l'envolupant de volum del tall gràcies a la línia que els uneix. Cada punt ve definit per la seva posició temporal dins del tall i el nivell de so que representa (des d'un màxim de 0 dB fins a un mínim de -40 dB).

Inicialment, hi ha un punt a l'inici i un altre al final del tall que es troben situats en el màxim nivell de so. Aquests dos punts no es poden eliminar i només se'n pot modificar el seu nivell.

Podem realitzar tres accions diferents amb el punts de control, sempre que l'opció Edit Envelope estigui activada:

- Afegir-ne: si cliquem amb el botó esquerre damunt de la representació de la forma d'ona, afegirem un nou punt de control a la posició i nivell on haguem situat el ratolí.
- Moure'ls: per modificar la posició o nivell d'un punt, cal que hi cliquem a sobre amb el botó esquerre i que el desplacem sense deixar anar el botó. L'únic que cal tenir en compte, és que un punt no es pot situar abans que el seu antecessor ni després que el seu predecessor.
- Eliminar-ne: per eliminar qualsevol punt de control que haguem afegit, li hem de fer doble clic. D'aquesta manera, la línia que representa l'envolupant es redibuixarà entre els dos punts més propers.

A continuació podem veure un tall amb l'envolupant modificada:

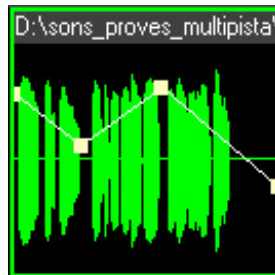


Fig 2.12 Tall amb l'envolupant modificada

CAPÍTOL 3. ARQUITECTURA

El DSI Visual Track ha estat desenvolupat amb el Visual Basic 6.0 perquè així s'indicava als requeriments inicials del projecte. De la mateixa manera, també era necessari dividir la programació del DSI Visual Track en dues parts ben diferenciades, el nucli (o kernel) i la interfície. Com veurem més endavant, finalment el programa ha quedat dividit en tres parts, una corresponent al nucli i dues a la interfície.

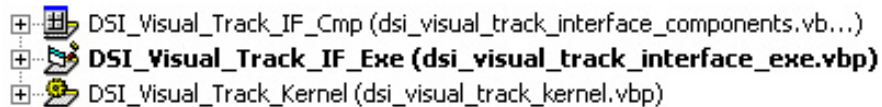


Fig 3.1 Grup de projectes que conformen el DSI Visual Track

3.1. El nucli

El nucli, que és el projecte anomenat “DSI_Visual_Track_Kernel”, és la part central del programa, la que s’encarrega de processar totes les operacions, ordres o dades que l’usuari li demani.

En el nostre cas l’hem creat en format de llibreria DLL ActiveX, ja que així podrà ser utilitzada de manera senzilla per qualsevol altra interfície que algú vulgui crear en un futur.

El nucli ha estat creat utilitzant la programació orientada a objectes, ja que ha estat dividit en classes. Aquestes classes són les que exposa la llibreria ActiveX per a poder-les utilitzar des de la interfície. Al primer apartat de l’annex hi ha la descripció completa perquè un programador pugui continuar amb el desenvolupament.

3.2. La interfície

La interfície és la part visible del programa, la que fa que l’usuari pugui treballar-hi d’una manera molt més fàcil i agradable a través d’una representació gràfica de tot allò que ofereix el nucli.

La nostra interfície es subdivideix en dues parts: els components i l’executable.

3.2.1. Els components

Els components de la interfície, projecte anomenat “DSI_Visual_Track_IF_Cmp”, queden englobats en un control ActiveX (extensió .ocx), el qual hem generat a partir de tots els controls d’usuari que hem ideat per representar els diferents elements del programa.

De la mateixa manera que hem comentat amb el nucli, així serà possible poder incorporar aquest control ActiveX dins d’un formulari d’un altre programa que el vulgui utilitzar, com ara el DSI Sound Station.

Els controls d’usuari que conformen el control ActiveX no són més que la representació gràfica de cada classe del nucli. D’aquesta manera, un usuari és capaç d’accedir a les funcions que ofereix una classe sense la necessitat d’haver de conèixer el seu funcionament intern, ja que és el control qui s’encarrega de fer totes les crides corresponents a la classe cada cop que l’usuari fa alguna acció.

3.2.2. L’executable

Per últim tenim la part executable de la interfície, el projecte anomenat “DSI_Visual_Track_IF_Exe”, que simplement és un formulari que conté el control d’usuari principal del nostre control ActiveX.

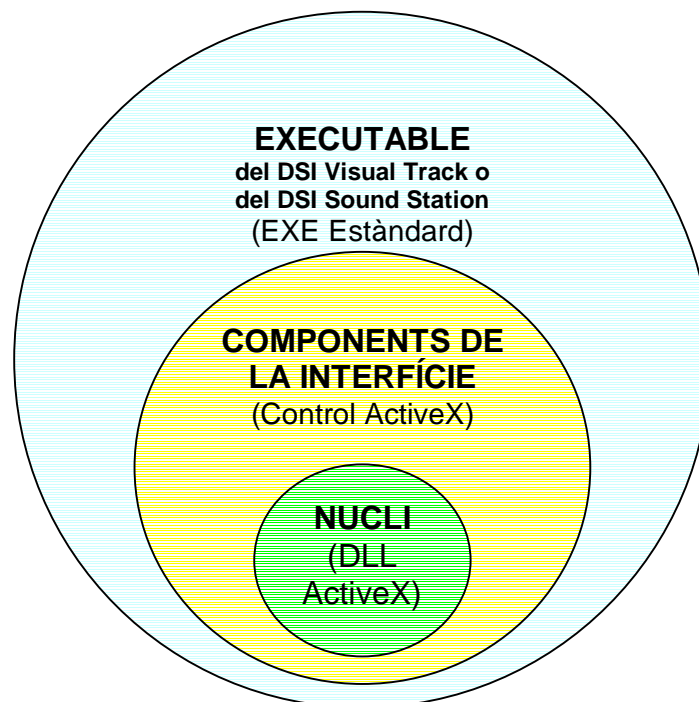


Fig 3.2 Esquema del DSI Visual Track

CAPÍTOL 4. FUNCIONAMENT INTERN DEL PROGRAMA

4.1. Kernel

4.1.1. Introducció

El kernel està compost per un mòdul (extensió .bas) i per una sèrie de mòduls de classe interrelacionats (extensió .cls). Totes les funcions que és capaç d'executar el programa es troben dintre dels mòduls de classes i generalment són cridades des de la interfície.

A l'hora de programar aquest nucli, vam tenir en compte que un altre usuari interessat en una estructura similar a la del nostre programa podria voler fer córrer el nucli a través d'una altra interfície. Per aquest motiu hem dissenyat el kernel de manera modular i utilitzant col·leccions d'objectes. D'aquesta manera hem intentat aconseguir un nucli que li doni una major flexibilitat a un futur programador interessat en el codi del DSI Visual Track. Un exemple d'aquesta propietat la tenim amb les pistes: el nucli suporta "n" pistes mentre que la interfície està limitada a només sis.

L'estructura d'arxius final del nucli té el següent aspecte:

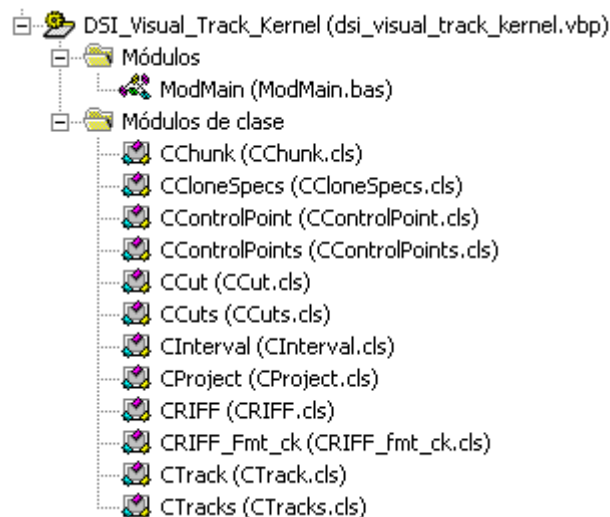


Fig 4.1 Components del kernel

El mòdul ModMain serveix exclusivament per allotjar les funcions que criden el "Logger" del DSI Sound Station i algunes altres que són necessàries per calcular l'energia dels talls.

En canvi, la potència real del kernel del multipista està a les classes.

Les classes estan ordenades jeràrquicament i agrupades de dues en dues allà on hi pot haver una col·lecció d'objectes.

- *CProject*: representa totes les propietats i les accions que es poden dur a terme sobre un projecte però que no tinguin res a veure amb les pistes ni els talls, com ara obrir o guardar un projecte.
- *CTracks / CTrack*: representa les pistes i tot el que tingui a veure amb propietats i funcions que afectin exclusivament a les pistes, com ara el temps inicial i final de reproducció.
- *CCuts / CCut*: representa els talls de so dins d'una pista i totes les propietats i funcions que els afecten, com ara afegir i eliminar talls.
- *CControlPoints / CControlPoint*: representa els punts de control dins d'un tall i tot el que és necessari per al seu funcionament, com les seves posicions i nivells.

De les classes esmentades, *CProject* i *CTracks* gaudeixen de les posicions més elevades, jeràrquicament parlant, cosa que podria induir a englobar-les en una sola classe que fes totes les seves funcions. Nosaltres hem decidit diferenciar-les en dues classes ja que conceptualment entenem millor que hi hagi unes funcions que únicament es dediquin a accions sobre el projecte en general, com ara controlar totes les accions que es fan i desfan sobre el projecte (classe *CProject*), i unes altres que únicament vagin adreçades a les pistes en si (classe *CTracks*).

En el següent diagrama podem veure l'ordre jeràrquic que segueixen les classes del kernel, començant per la classe que engloba totes les altres i acabant per les que tenen menys pes dins de la jerarquia de classes.

A més, hi ha unes classes addicionals (*CRIFF*, *CRIFF_Fmt_ck*, *CChunk*, *CCloneSpecs* i *CInterval*) que estan agafades del DSI Sound Station original i serveixen per manipular arxius.

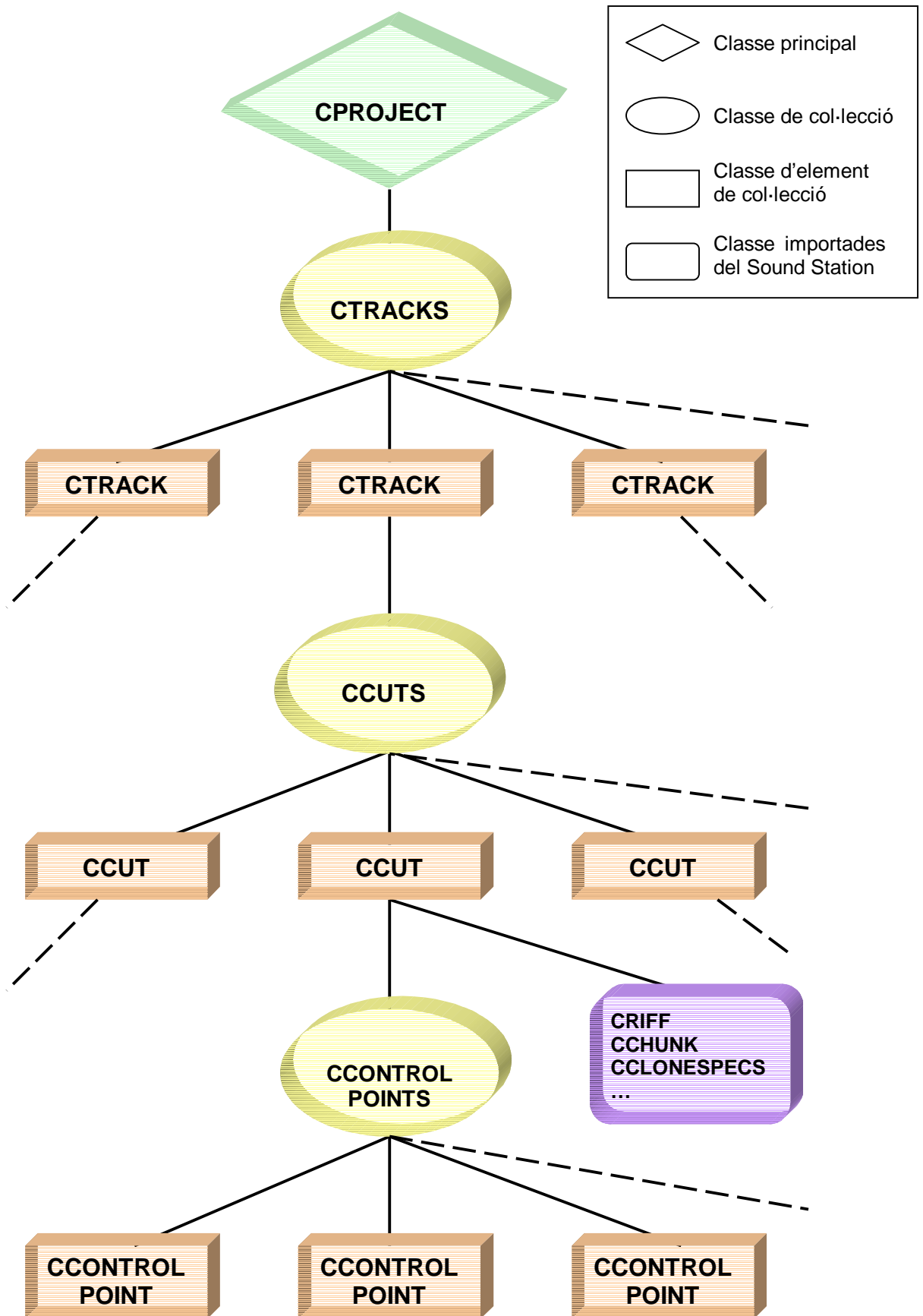


Fig 4.2 Arbre jeràrquic de les classes del kernel

4.1.2. DirectShow Editing Services

Per poder aconseguir que es reproduís un arxiu MPEG-2 vam aprofitar alguna de les funcions que havia creat un projectista (José Molina), que també havia fet el TFC per a l'empresa Digital Sound Items. Això ens va servir per entrar en les diferents funcionalitats que ofereix DirectX i totes les seves respectives API's (Application Programming Interface).

En un principi la idea per fer una reproducció de múltiples talls de so la basàvem en anar "disparant" els talls a mesura que la seva posició inicial anava coincidint amb un clock de reproducció. Encara que soni senzill, la implementació comportava certes dificultats de programació, i paral·lelament vam buscar altres mètodes dintre de les variades llibreries dedicades a dades multimèdia que ofereix el DirectX.

Curiosament l'API que més es va adequar a les nostres necessitats era una que estava enfocada cap a edició de vídeo. Aquesta aplicació de programació se la coneix com a DirectShow Editing Services (DES) i permet fer una abstracció de DirectShow, dins la complexitat que aquest comporta, oferint a l'usuari una sèrie d'interfícies dissenyades exclusivament per manipular vídeo i el seu àudio associat.

El nucli de DirectShow té una arquitectura suficientment robusta com per suportar diferents formats d'arxius multimèdia sense que el programador s'hagi de preocupar de detalls com la compressió o el Sample Rate per fer la lectura d'un arxiu. El problema arriba quan l'edició multimèdia que es vol realitzar no és lineal, per exemple: es necessita crear un projecte on es requereixen els últims tres segons de música d'un mp3 mentre en aquest interval han de començar a reproduir-se dos sons més.

Aquests problemes de reproduccions simultànies o fins i tot parcials es poden solucionar amb molta menys dificultat utilitzant l'anteriorment esmentat DES. A efectes de facilitar la comprensió de la interacció que resulta entre un aplicació creada per un programador i el nucli de DirectShow adjuntem el següent esquema explicatiu:

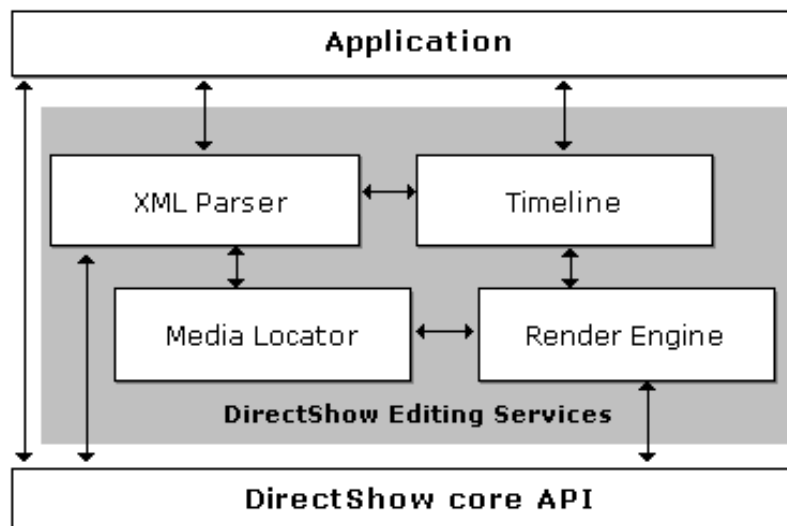


Fig 4.3 Interacció DirectShow – Aplicació d'usuari

La *Timeline*: Com el seu nom indica és principalment una línia de temps, que conté les col·leccions de talls, transicions entre elements i efectes del projecte organitzats en pistes.

Render Engine: És la part que s'encarrega de fer les reproduccions (previews com les anomena el Directshow) o renderitzar el projecte generant una sortida d'un sol arxiu d'àudio.

Media Locator: Manté en memòria les ubicacions al disc dur dels arxius multimèdia que s'utilitzin al projecte.

XML Parser: Escaneja la Timeline per crear un arxiu de sortida en format XML per mantenir la informació del projecte. També permet fer el procés invers: d'un arxiu XML pot crear una Timeline amb la informació emmagatzemada.

Dels elements esmentats només farem una referència sobre la Timeline degut a l'abstracció del concepte.

4.1.2.1. El model de Timeline

La Timeline és un objecte que utilitza el DirectShow Editing Services per tal de representar tot el que conté un projecte d'edició d'àudio. Encara que el DES està enfocat a l'edició de vídeo, a partir d'ara només farem esment de les funcionalitats d'àudio d'aquest.

Un projecte d'edició comporta un conjunt d'arxius d'àudio (sources) que es poden inserir seqüencialment en una pista (track). La versatilitat del DES radica en la possibilitat de poder tenir altres pistes amb altres muntatges en seqüència diferents i poder-les mesclar i fins i tot incloure efectes com reverberacions o transicions com els Fades entre elements. Com en el nostre propi projecte, una

Timeline té una estructura jeràrquica en forma d'arbre, però en aquest cas l'element principal és el propi objecte Timeline, com mostra la següent figura:

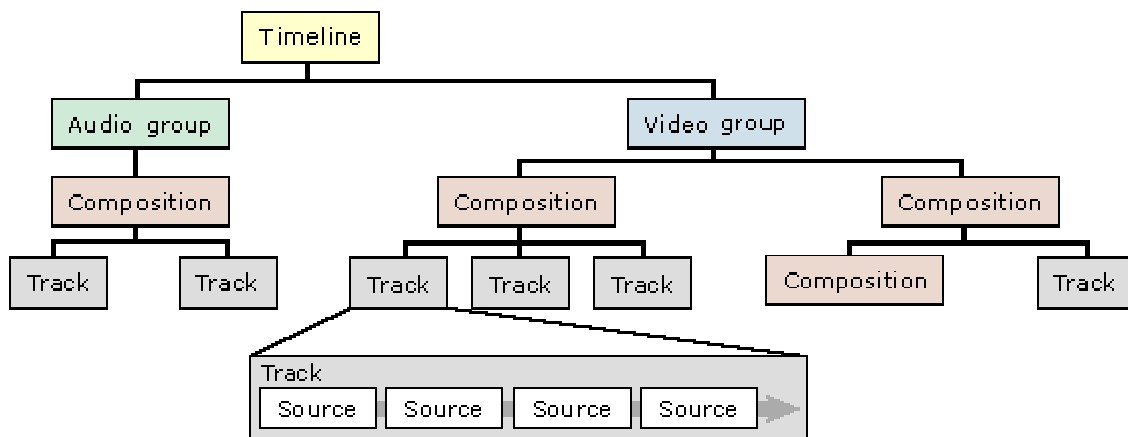


Fig 4.4 Relació jeràrquica dels elements d'una Timeline

Gràcies a aquest paral·lelisme entre l'estructura de les nostres classes del nucli i la Timeline la reproducció d'un projecte amb el nostre programa s'ha simplificat força. Així cada cop que es fa una crida de reproducció es genera un objecte Timeline que conté un Group d'àudio, que deriva en una Composition exclusivament d'àudio que inclou el mateix número de pistes amb les que s'estava treballant i se li donen a cada pista els noms dels talls corresponents amb les seves posicions i intervals. El resultat final de la reproducció podria ser similar al del següent esquema temporal:

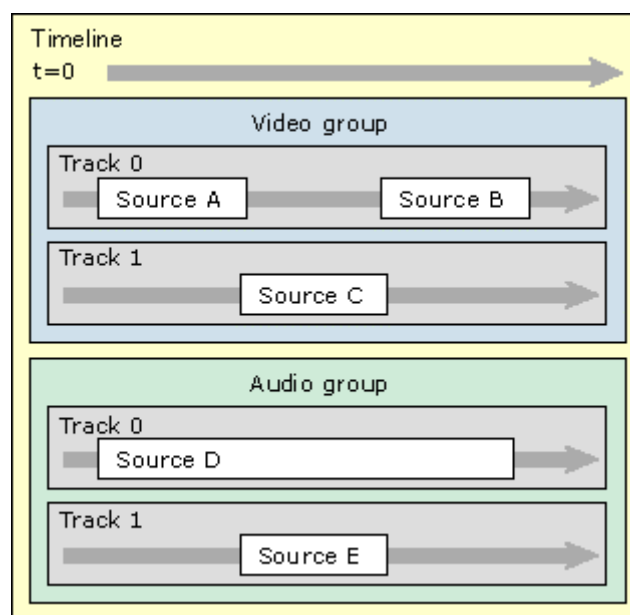


Fig 4.5 Esquema temporal de reproducció

4.1.3. Aspectes i funcions destacables del kernel

OpenProject i SaveProject (OpenTL i SaveTL)

En una primera instància vam començar a programar aquestes funcions manualment, es a dir, nosaltres mateixos generàvem les funcions adequades per escanejar totes les classes i col·leccions i així extreure la informació de cada item. Seguidament aquestes dades les escrivíem en un arxiu forçant una estructura de tipus XML.

Actualment el programa utilitza una funcionalitat de DirectShow que és capaç de generar l'arxiu XML a partir de la Timeline i viceversa. L'objecte que s'encarrega de realitzar aquestes accions és el Xml2Dex i les dues funcions utilitzades són ReadXMLFile i WriteXMLFile, a les quals només cal passar-los la Timeline creada i la ruta de l'arxiu.

Undo i Redo

Per fer aquestes funcions del programa vam haver de decidir quina tècnica utilitzàvem entre les tres que ens van semblar factibles d'implementar.

La primera opció consistia en què l'usuari generés un arxiu backup cada cop que considerés que podria necessitar aquell estat del projecte. La part negativa d'aquest mètode era la impossibilitat de fer més d'una acció Undo.

La segona opció era generar un mètode que emmagatzemés les accions efectuades sobre el projecte per, a l'hora de fer un Undo, cridar una ordre que fos la inversa de l'efectuada per l'usuari. El problema amb el que ens vam trobar era l'haver de crear una funció específica per cadascuna de les que ja teníem programades, amb la qual cosa el volum de codi creixeria considerablement.

La tercera opció va resultar ser la definitiva. Aprofitant el Parser d'Xml que ofería el DirectShow Editing Services vam decidir que a cada acció realitzada generàvem un arxiu temporal en XML igual als utilitzats per guardar els projectes. D'aquesta manera no generàvem un pes excessiu al crear els arxius temporals i érem capaços de desfer i refer totes les accions. Així per carregar una acció de desfer o refer només calia utilitzar la funció OpenTL (funció privada cridada per OpenProject) que hem comentat abans, i per crear un arxiu temporal vam escriure la funció SaveTemp, que utilitza SaveTL (funció privada que utilitza SaveProject) per generar un arxiu amb el nom VTTEMPX.xml, on la X representa el número d'operació.

Paral·lelament existeix una propietat anomenada OperationCount que ens permet portar el compte de les operacions realitzades i informar-nos del número de l'operació actual.

Edited

Aquesta propietat Boolean va lligada a la funció SaveProject. La seva missió és permetre'ns saber en quina operació s'ha guardat el projecte. D'aquesta manera el programa decideix quan ha de preguntar a l'usuari si vol salvar el projecte a l'hora de tancar el programa.

MaxLength

S'encarrega d'esbrinar en tot moment quina és la durada màxima del projecte. Així, cada cop que s'ha afegit, mogut, seleccionat un interval o eliminat un tall es desencadena una sèrie d'esdeveniments per actualitzar aquesta propietat, tenint en compte la posició dels talls a totes les pistes.

Gràcies a MaxLength podem, per exemple, saber el temps final que haurem d'utilitzar a l'hora de reproduir un projecte.

Play (PlayTL)

Aquesta és la funció principal de reproducció. S'encarrega de decidir quin serà l'interval que es reproduirà depenent de:

- La posició de la barra d'inici
- L'existència d'un tram seleccionat
- El mode de reproducció escollit

Un cop determinades totes les variables anteriors es genera una Timeline específica per aquella reproducció. A partir d'aquí existeixen dues variants: que la reproducció sigui contínua o que hagi d'efectuar un salt en el temps (saltar-se el tram seleccionat). Depenent del cas es crearà una Timeline contínua o es recalcularan els temps interns de la Timeline perquè no hi hagi buits de so durant el salt.

CreateTimeline

Aquesta funció és la que s'utilitza per omplir la Timeline amb el nombre adequat de pistes i els talls inserits en aquestes, a partir de les dades contingudes a les classes. Un cop la Timeline està plena, podem dir que es troba preparada per ser reproduïda o guardada en XML.

CreateFromTimeline

Aquesta funció es pot considerar com la inversa de l'anterior. La seva funció és omplir les classes del nucli a partir de les dades extretes d'una Timeline. Aquesta és una de les funcions principals que s'utilitzen a l'hora de recuperar un arxiu XML i és cridada des d'OpenProject.

Procediments Friend dintre de les classes de col·lecció

Una part important del funcionament del programa es basa en el sistema d'esdeveniments de Visual Basic, que permet executar un procediment en una classe, llançant un esdeveniment des d'una altra.

Pel fet d'utilitzar col·leccions d'objectes en les classes CTracks, CCuts i CControlPoints ens vam trobar amb una limitació de Visual Basic: des de les classes que contenen una col·lecció no es poden rebre esdeveniments llançats des dels seus elements.

La solució consisteix en definir uns procediments de tipus Friend (que només són visibles des del seu projecte) a les classes de col·lecció que fan el paper dels procediments que s'executen quan es rep un esdeveniment. Des de les classes d'element només cal fer la crida d'aquests procediments tal com es faria amb un esdeveniment, però abans cal haver definit la classe que conté els elements.

Per exemple, des de CTrack es pot cridar un procediment de CTracks de la següent manera, sempre i quan haguem definit la variable *iTracks* com la classe que conté la col·lecció de pistes: *iTracks.AddedCut*

ValidatePosition i AreCompatible

Aquest parell de funcions són la pedra angular de la inserció i desplaçament dels talls a les pistes. L'AreCompatible és cridat pel ValidatePosition i s'encarrega de comprovar si dos talls se superposen entre si, mentre que el ValidatePosition escaneja la col·lecció de talls per anar-los comparant un a un amb el que es vol afegir o moure.

La condició de compatibilitat entre dos talls A i B és que (inici d'A \geq final de B) OR (final d'A \leq inici de B).

Propietats d'un tall relatives al temps

Quan inserim un tall en una pista, la seva posició en el temps ve definida per 4 propietats. Per una banda tenim StartTime i EndTime, que defineixen la posició del tall respecte l'eix temporal de la pista. I per l'altra, tenim StartInterval i EndInterval, inici i fi de la secció a reproduir del tall respecte el seu propi eix temporal. Per tant al modificar alguna d'aquestes últimes propietats, els valors d'StartTime o EndTime s'adaptaran automàticament a la nova durada del tall.

GetEnergy

Aquesta funció s'encarrega de trobar els nivells d'energia d'un tall que posteriorment s'utilitzaran per crear un dibuix aproximat de la seva forma d'ona. Els passos que segueix són els següents:

- 1- A partir d'una avaluació inicial que ens indica si aquest arxiu es de tipus Riff/Wav (MPEG- LayerII), utilitzem el nombre de bytes que conté l'arxiu per saber quants paquets de dades el componen.
- 2- Omplim una classe anomenada CChunk que ens permetrà moure'ns per dintre d'un paquet de dades.
- 3- Situem el "punter de lectura" just a l'inici de les dades de l'energia del so, gràcies a un càlcul previ de l'Offset de bytes que existeix abans de trobar-nos amb les dades que ens interessin.
- 4- Llegim els nivells d'energia
- 5- Transformem les dades a dB's.
- 6- Anem fent un bucle a través dels diversos paquets que contenen dades de l'energia de l'ona per llegir-los tots.

Nivells utilitzats per definir els punts de control

Els punts de control que hem creat dintre dels talls vénen definits per la seva posició horitzontal (temps) i vertical (en el nostre cas dB's que representen el nivell de volum).

Nosaltres hem definit el nivell màxim com a 0 dB (volum nominal del tall) i el mínim com a -40 dB .

Però aquesta representació vertical que nosaltres hem utilitzat, només és una de les diverses aplicacions que se'ls pot donar. Una altra pot ser controlar el panorama (balanç entre els canals dret i esquerre). En aquest cas s'haurien de definir uns altres marges, per exemple: -1 per tota la potència derivada al canal esquerre i +1 al dret.

ValidateControlPoint

Tal com passava amb els talls, els moviments dels punts de control també s'han de controlar. A l'hora de moure un punt, s'ha de tenir un compte que aquest no se surti dels quatre marges del propi tall i que no se'n vagi a la dreta o esquerra (moviments en l'eix temporal) dels punts anterior i posterior.

D'entre tots aquests casos, n'hi ha un que va resultar més complicat de controlar: vigilar que el punt no se sortís per la dreta del tall. Això és degut a que el valor que defineix on s'acaba un tall (la seva durada en temps) només és conegut pel propi tall, però no per la seva col·lecció de punts.

Per solucionar-ho, vam haver de programar un esdeveniment que s'executava cada cop que un punt es movia i que era rebut per la classe CCut. Allà, es

podia comprovar fàcilment si s'estava intentant situar un punt de control fora del límit dret del tall, i impedir-ho.

4.2. Interfície

4.2.1. Introducció

La interfície està integrada per dos projectes diferents dins del nostre grup de projectes del Visual Basic.

El primer està compost per un mòdul (extensió .bas) i per una sèrie de controls d'usuari (extensió .ctl). Els controls d'usuari són els encarregats d'interaccionar amb les classes del nucli, cadascun amb la classe que representi a la interfície. Aquest primer bloc l'hem anomenat "Components de la interfície", i la llista dels seus arxius és la següent:

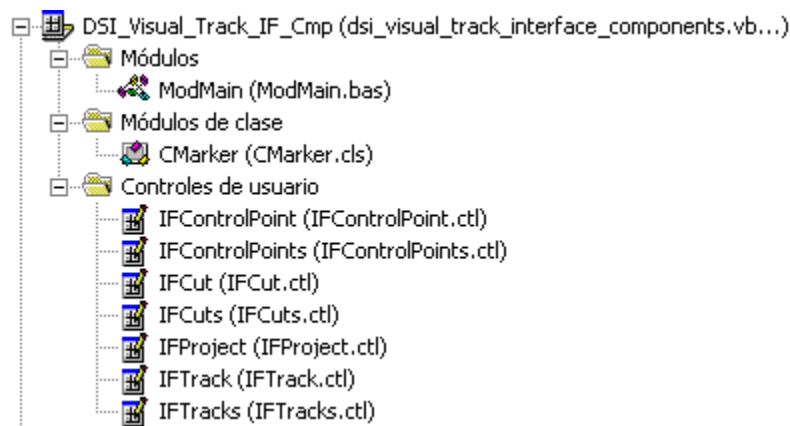


Fig 4.6 Components de la interfície

El mòdul ModMain serveix principalment per allotjar les funcions que criden el "Logger" del DSI Sound Station.

Els controls, per la seva banda, es corresponen un a un amb cada classe del nucli. Per tant, tot el que hem comentat sobre l'ordre jeràrquic de les classes i la seva distribució en objectes i col·leccions d'objectes, també descriu l'organització dels controls d'usuari.

Per altra banda, tenim el projecte que hem anomenat "Executable de la interfície", que està format per un mòdul (extensió .bas) i per dos formularis (extensió .frm). El formulari principal (FrmMain) representa l'executable principal del programa, i tal com hem fet nosaltres, qualsevol altra persona que desitgi incorporar els nostre editor multipista dins d'un altre programa, simplement hauria d'agafar el control IFProject i inserir-lo en el formulari corresponent.

Aquest és l'aspecte del projecte que conté el formulari:



Fig 4.7 Executable de la interfície

En els següents diagrames podem veure com queden estructurats els diferents elements de la interfície, començant pel formulari que ho engloba tot i acabant pels elements més interns, és a dir, la sèrie de controls IFControlPoint.

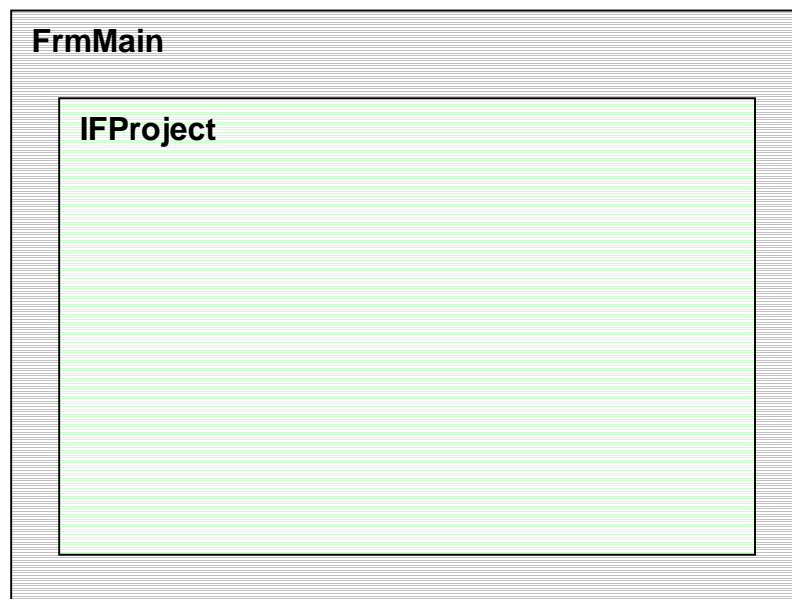
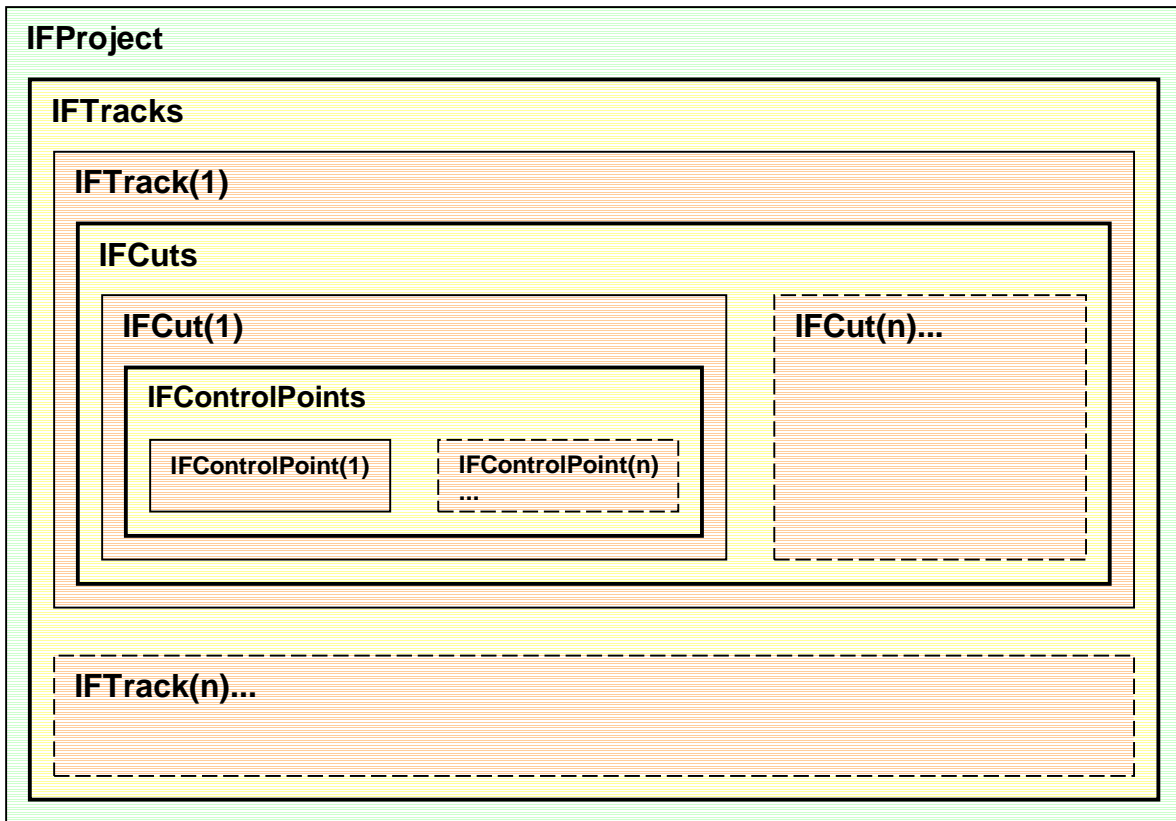


Fig 4.8 Esquema del formulari (executable) amb el control d'usuari principal del control ActiveX



- Control de col·lecció
- Control d'objecte
- Següents controls d'objecte

Fig 4.9 Esquema dels control d'usuari del control ActiveX

4.2.2. Aspectes i funcions destacables de la interfície

InitialNumTracks

En un principi, abans de dividir la interfície en dos projectes (executable i components), dintre d'IFProject teníem prefixat que cada projecte nou s'iniciaria amb 4 pistes buides.

Quan vam separar l'executable, vam creure oportú que no fos IFProject qui decidís el nombre inicial de pistes, sinó que fos l'executable qui imposés quantes en volia.

D'aquesta manera, qualsevol executable com el del DSI Sound Station, és capaç de controlar el nombre de pistes amb les quals vol treballar, i fins i tot pot utilitzar el DSI Visual Track com un editor de so monopista.

Representació de les pistes

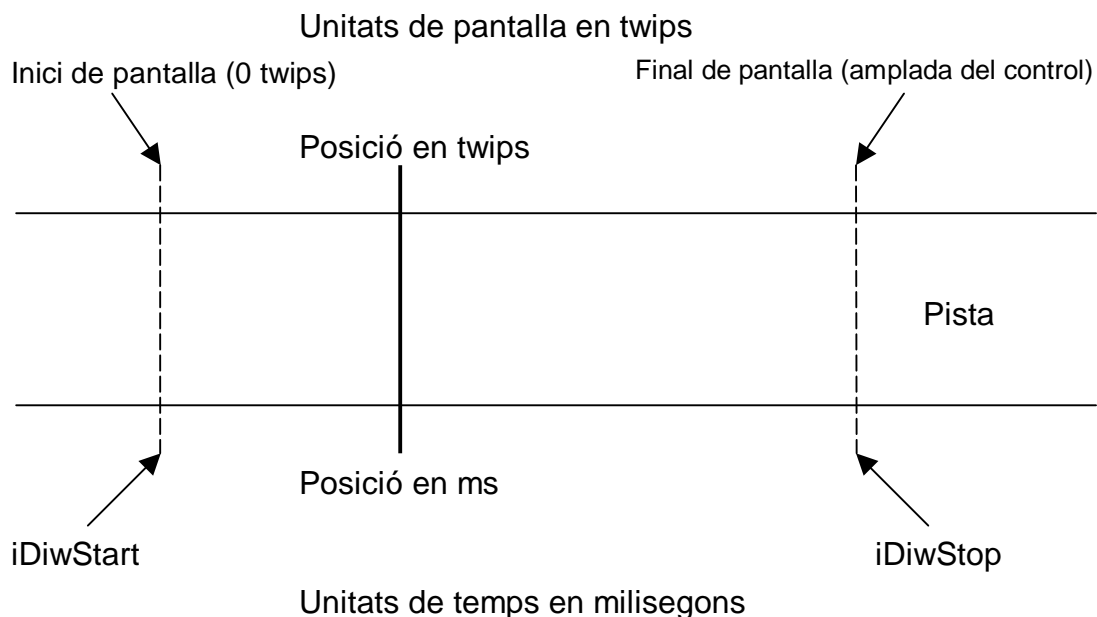


Fig 4.10 Correspondència entre unitats de temps i de pantalla

Aquest diagrama mostra el concepte en el qual ens hem basat per fer la representació de les pistes, que es basa en una correspondència entre l'amplada de les pistes i el rang de temps que representa. Aquesta relació entre milisegons i twips és calculada per la propietat *TimeScale* de la següent manera:

$$TimeScale = \left(\frac{iDiwStop - iDiwStart}{\text{amplada del control}} \right) \quad (4.1)$$

$iDiwStart$ i $iDiwStop$ representen l'inici i final de la finestra visible de les pistes en milisegons, mentre que l'amplada és un valor fix en twips. Les unitats del factor $TimeScale$ són per tant: nombre de milisegons per twip.

Aquesta finestra que emmarquen $iDiwStart$ i $iDiwStop$ és la que ens permet moure'ns a través de la llargada de les pistes. Existeixen dues maneres de modificar-la:

- 1- Utilitzant el zoom. En aquest cas modifiquem el factor $TimeScale$ de manera que la finestra visible augmenta o es redueix.
- 2- Utilitzant la barra de desplaçament. D'aquesta manera no modifiquem l'amplada de la finestra visible sinó que la desplaçem al llarg de la pista (variem $iDiwStart$ i $iDiwStop$ però mantenim la seva proporcionalitat).

Per últim destacarem un parell de funcions:

$MsToDisplayUnits$, que serveix per passar una posició en milisegons provinent del nucli a la corresponent posició en pantalla tenint en compte la finestra visible.

$$\text{posició en pantalla} = (\text{posició en ms} - iDiwStart) \times \left(\frac{\text{amplada del control}}{iDiwStop - iDiwStart} \right) \quad (4.2)$$

$SetDiwStartStop$, que apart de servir per fixar els valors de la finestra, actualitza les posicions de la barra d'inici, d'una possible selecció i dels elements de totes les pistes.

Comunicació nucli-interfície

Aquest és un dels punts més rellevants del funcionament del programa. Es tracta de l'ús del sistema d'esdeveniments del nucli cap a la interfície, de manera que si un usuari accedeix al nucli directament (per exemple, en mode consola) qualsevol canvi que pugui introduir també es veuria reflectit a la interfície.

Perquè el lector es pugui fer una idea d'aquest mecanisme, l'explicarem utilitzant el següent exemple referit a la inserció d'un tall, tot i que qualsevol altra acció des de la interfície generaria una cadena d'esdeveniments similar.

Quan volem afegir un tall des de la interfície l'arrosseguem fins a la posició desitjada dins de la pista. Just després de deixar-lo anar, la interfície crida a la funció corresponent del nucli per afegir un tall passant-li totes les variables necessàries. Seguidament el nucli fa les comprovacions necessàries i decideix si es pot afegir o no. Si l'ha afegit correctament llança un esdeveniment cap a la interfície per tal que aquesta afegeixi un nou control d'usuari que representi aquell tall i posteriorment pintar-lo en pantalla. En canvi, si el nucli no ha

aconseguit afegir-lo no llança cap esdeveniment i la interfície no executa cap acció.

Funcionament dels desplaçaments i seleccions

Tots aquests tipus de moviments (desplaçament de talls i punts de control i selecció a les pistes) es fonamenten en el sistema d'esdeveniments que es generen en utilitzar el ratolí.

Per començar, hem de recollir l'esdeveniment `MouseDown` sobre el control corresponent. En aquest punt hem de guardar en una variable boolean l'estat de l'acció. Per exemple, en el cas de realitzar una selecció, posem la nostra variable `Selecting` a `True`. A partir d'aquí, rebrem continuadament l'esdeveniment `MouseMove` mentre movem el ratolí per sobre del control, i com `Selecting` està activat, cridarem la funció corresponent del nucli per informar-lo d'aquesta acció. Una vegada es generi l'esdeveniment `MouseUp` per haver deixat de prémer el botó principal del mouse, canviarem `Selecting` a `False` per tal de deixar de generar esdeveniments procedents d'aquell control i així finalitzar l'acció.

GrabWaveForm

Aquesta funció es troba lligada íntimament amb la funció `GetEnergy` prèviament explicada al nucli. D'una banda pinta físicament l'ona punt a punt a partir de les dades que li ofereix `GetEnergy`. Però d'altra banda és la que s'encarrega que l'ona es pugui veure juntament amb els punts de control i no es tapin mútuament.

El problema estava en què el control que recull els valors de `GetEnergy` és `IFCut`, però aquest control és invisible perquè queda tapat per `IFControlPoints`. La solució va ser omplir la propietat `Picture` d'`IFControlPoints` utilitzant una captura de la forma d'ona agafada del control `IFCut`. D'aquesta manera la forma d'ona queda com a "wallpaper" d'`IFControlPoints` i l'usuari pot interaccionar lliurement amb els punts de control.

CONCLUSIONS I PROPOSTES FUTURES

Aquest projecte es va iniciar el febrer de 2005 amb la idea de que fos presentat al juny, però donada l'envergadura del projecte i els limitats coneixements de Visual Basic que teníem a l'inici, finalment vam decidir presentar-lo al setembre. El resultat però, ha estat molt satisfactori. Durant els cinc mesos i escaig que hem estat programant hi ha hagut una evolució notable en la manera d'entendre la programació en Basic. Començant per coses tan bàsiques com el funcionament dels esdeveniments en el Visual Basic fins a d'altres més complexes com acabar integrant funcionalitats de DirectX utilitzant algunes de les API's que ofereix.

Un punt que ha estat clau per la realització del DSI Visual Track ha estat el bon ritme de treball que s'ha mantingut durant el transcurs de "gestació" del programa, amb una mitjana d'unes 3'5 hores diàries i buscant un dia a la setmana per poder veure'ns amb el tutor (normalment al cap de setmana). Buscar un punt de treball on poder posar en comú el treball realitzat personalment i on poder treballar conjuntament ha facilitat també la bona evolució del treball.

Val a dir que al projecte no hi han mancat les complicacions. Una de les parts amb les que hem tingut més dificultats ha estat la part de reproducció i renderitzat del programa. No tant per la complexitat dels conceptes en utilitzar les llibreries de DirectX sinó per les limitacions que DirectShow té amb el llenguatge Visual Basic. Cal tenir en compte que un dels requeriments d'aquest projecte era l'ús de Visual Basic per la programació, però segurament un llenguatge més estès com C++ o més nou com C# hauria facilitat alguns aspectes, ja que una de les traves que ens vam trobar era la falta de documentació que existia per a Visual Basic.

Les dues limitacions més importants han estat, per una banda, el no poder programar la secció de renderitzat del projecte, i per altra banda, la creació d'efectes, que en el nostre cas hauria servit per la definició d'una envolupant per controlar el volum instantani. En la majoria de casos els exemples de codi que trobem a l'MSDN són per a C++ així que, encara que Visual Basic també suporti la llibreria, el programador ha de fer un esforç afegit de "traducció".

En el cas de renderització del projecte vam arribar a un punt on, per sorpresa nostra, vam veure que una de les interfícies de DirectShow no existia per a Visual Basic. Concretament una anomenada ICaptureGraphBuilder2 que és la que s'encarrega d'escriure el projecte en un sol arxiu, característica fonamental en un programa d'àudio. Això vol dir que tècnicament és impossible crear un arxiu de so a partir d'una Timeline amb el Visual Basic.

Amb la part dels efectes ens va succeir una cosa semblant: Vam aconseguir programar l'entorn adequat per afegir l'efecte però a l'hora d'afegir-lo ens vam trobar amb que el Visual Basic no era capaç de reconèixer la cadena de caràcters que l'identifiquen. Per tal que el lector es faci una idea de les mancances de compatibilitat que tenen el Visual Basic i DirectShow adjuntem a

l'apartat 3 de l'annex una taula extreta directament de l'MSDN (pàgina de Microsoft d'ajuda als desenvolupadors) amb la llista de mètodes de l'objecte AMTimelineObj.

Com es pot apreciar a la columna de descripció de mètodes hi ha una gran quantitat de funcions que Visual Basic no suporta. Alguns d'aquests mètodes se supleixen amb els que tenen el seu mateix nom més un 2 o les sigles VB al final, però aquests casos són minoritaris en la resta d'objectes que treballen amb entorns de Timeline.

Veient aquestes limitacions només hi ha una sortida possible si es vol focalitzar el sistema de reproducció del programa utilitzant DirectShow: crear una llibreria DLL externa al nucli programada amb C++ o C#.

És en aquest punt on fonamentem les propostes futures del programa.

Veient que totes les altres funcionalitats bàsiques del programa estan assolides només quedaria solucionar els dos problemes prèviament esmentats (renderitzat i efectes d'envolupant); així que en un projecte futur es podria crear una llibreria en C que es pogués annexionar al nucli perquè les seves funcions fossin cridades des d'aquest. D'aquesta manera en utilitzar C no ens trobaríem limitats en l'accessibilitat dels mètodes i es podria gaudir de l'àmplia documentació que existeix per a aquest llenguatge, tant en fòrums o pàgines dedicades d'Internet com a l'MSDN, on abunden els exemples de codi en contraposició amb el que succeeix amb Visual Basic.

Per últim, i deixant a part les propostes futures, podem dir que estem força satisfets amb la realització d'aquest programa, perquè el resultat ha sigut molt proper a la idea inicial que teníem d'un reproductor multipista. Tanmateix, el que més ens satisfà és saber que el programa, quan arribi a la versió 1.0, farà un bon servei als professionals de Ràdio Sant Cugat.

BIBLIOGRAFIA

- [1] Ceballos Sierra, F.J., *Enciclopedia de Microsoft® Visual Basic™ 6*, Ra-Ma Editorial, Paracuellos de Jarama (Madrid), 1999.
- [2] <http://msdn.microsoft.com/library>, *Microsoft Developer Network (MSDN) Library*
- [3] <http://www.vbaccelerator.com/home/VB/index.asp>, *Secció de Visual Basic de la pàgina Visual Basic Accelerator.*

ANNEXOS

1. Descripció de les classes del nucli

En els següents paràgrafs veurem una breu descripció del funcionament intern del kernel, les classes amb les seves respectives col·leccions i les funcions més importants que aquestes contenen.

CProject.cls

Aquesta classe és la que engloba totes les altres i es diu així perquè conté tots els elements que conformen un projecte creat amb el DSI Visual Track.

L'element més important de tots els que conté és CTracks, ja que és el que representa el següent nivell en l'ordre jeràrquic de les classes.

- Atributs privats:

- o Const **ClassName** As String = "CProject"
- o Const **Drive** As String = "C:\"
- o Const **Prefix** As String = "VTTEMP"
- o Const **ProjectName** As String = "DSI_Visual_Track"
- o Const **TempDir** As String = "VTTEMP"
- o **Control** As IMediaControl
- o **Graph** As IGraphBuilder
- o **iName** As String
- o **iOperationCount** As Integer
- o WithEvents **iTracks** As CTracks
- o **MaxOperationCount** As Integer
- o **MyEvent** As IMediaEvent
- o **PlayingSplitted** As Boolean
- o **Position** As IMediaPosition
- o **R** As RenderEngine
- o **SavedOperationCount** As Integer
- o **Split1** As Long
- o **Split2** As Long
- o **T** As New AMTimeline
- o **Timeunit** As e_Timeunit

- Membres públics:
 - o Funcions i procediments:
 - **Class_Initialize** ()
 - **Class_Terminate** ()
 - **DeleteAllTempFiles** (): elimina tots els arxius temporals.
 - **NewProject** (): reinicia totes les variables per poder començar un nou projecte.
 - **OpenProject** (ByVal FileName As String): obre un projecte desat en format XML.
 - **Play** (ByVal FromMs As Long, ByVal UntilMs As Long, Optional Splitted As Boolean = False): reproduïx el projecte que es trobi obert. S'han de passar com a paràmetres els temps inicial i final de reproducció en milisegons i, opcionalment, s'ha d'indicar si el que es vol reproduir és l'interval no seleccionat del projecte.
 - **Redo** (): refà l'última acció prèviament desfeta.
 - **Render** (): no implementada
 - **SaveProject** (ByVal FileName As String): desa un projecte en format XML.
 - **SaveTemp** (): desa un arxiu temporal en format XML per així poder utilitzar les funcions d'*Undo* i *Redo* (desfer i refer).
 - **StopPlaying** (): atura una reproducció en curs.
 - **Undo** (): desfà l'última acció.
 - o Propietats:
 - Get **Edited** () As Boolean: indica si un projecte ha estat modificat després d'haver estat guardat.
 - Get **MaxLength** () As Long: durada màxima d'un projecte en milisegons.
 - Get **Name** () As String: nom amb el que ha estat guardat un projecte.
 - Get/Let **OperationCount** () As Long: número d'operació del projecte en el què ens trobem.

- **Get *Playing*** () As Boolean: indica si hi ha cap reproducció en curs.
 - **Get *PlayPosition*** () As Long: retorna el temps actual de reproducció en milisegons.
 - **Get *Tracks*** () As CTracks: retorna la col·lecció de pistes de tipus CTracks.
- Esdeveniments:
- **AddedCTracks** (): no utilitzat.
 - **AddedCut** (NewCut As CCut): s'ha afegit un tall.
 - **AddedOrMovedControlPoint** (ControlPoint As CControlPoint): s'ha afegit o modificat un punt de control.
 - **ChangedCut** (Cut As CCut): s'ha mogut un tall.
 - **ChangedMaxLength** (NewLength As Long): la durada del projecte ha variat.
 - **EnableUndoButton** (State As Boolean): indica si és possible desfer una acció.
 - **EnableRedoButton** (State As Boolean): indica si és possible refer una acció.
 - **NewCutInterval** (Cut As CCut): s'ha modificat l'interval reproduïble d'un tall.
 - **OpenedProject** (): no utilitzat.
 - **RemovedControlPoint** (index As Integer): s'ha eliminat un punt de control.
 - **RemovedCut** (index As Integer): s'ha eliminat un tall.
 - **RemovedTrack** (index As Integer): s'ha eliminat una pista.
- Membres privats:
- Funcions i procediments:
 - **CreateCutInNotSplittedTimeLine** (ByRef Cut As CCut, ByRef TrackTL As AMTimelineTrack): afegeix un tall de so a la *Timeline*, tenint en compte que la reproducció serà contínua.
 - **CreateCutInSplitTimeLine** (ByRef Cut As CCut, ByRef TrackTL As AMTimelineTrack): afegeix un tall de so a la

Timeline, tenint en compte que la reproducció no serà contínua.

- **CreateCutInTimeLine** (ByRef Cut As CCut, ByRef TrackTL As AMTimelineTrack, StartTime As Long, EndTime As Long, StartInterval As Long, EndInterval As Long): afegeix un tall a la *Timeline*.
- **CreateFromTimeline** (): omple les classes del nucli a partir d'una *Timeline*.
- **CreateTimeline** (): omple una *Timeline* a partir de les classes del nucli, sense tenir en compte l'estat de les variables *mute* i *solo* de cada pista.
- **CreateTimeLineForPlay** (Splitted As Boolean) omple una *Timeline* a partir de les classes del nucli, tenint en compte l'estat de les variables *mute* i *solo* de cada pista.
- **CreateVoidTrackInTimeLine** (ByRef TrackTL As AMTimelineTrack, ByRef GCompo As AMTimelineComp): afegeix una pista a la *Timeline*.
- **FillTrackInComposition** (ByRef Track As CTrack, ByRef GCompo As AMTimelineComp, Splitted As Boolean): omple una pista de la *Timeline* amb els talls de so corresponents segons les classes del nucli.
- **OpenTL** (ByVal FileName As String) llegeix un arxiu XML que prèviament hagi estat creat a partir d'una *Timeline*.
- **PlayTL** (FromMs As Long, UntilMs As Long): reproduïx la *Timeline* amb l'inici i finals indicats en milisegons.
- **RenderTL** (): no implementada
- **SaveTL** (ByVal FileName As String): crea un arxiu XML amb informació sobre l'estat actual de la *Timeline*.
- **SetAudioGroupInTimeline** () As AMTimelineComp: estableix el tipus de grup a la *Timeline*.
- **StopTL** (): atura la reproducció de la *Timeline*.
- **TempDirName** (num As Integer): retorna la localització del directori temporal.

CTracks.cls

Aquesta classe fa referència a tot el que engloba les pistes. Principalment conté una col·lecció d'elements pista (ctrack) tal i com s'explica a la següent classe.

- Atributs privats:
 - o Const **ClassName** As String = "CTracks"
 - o **iCollection** As New Collection: col·lecció d'elements CTrack (pistes)
 - o **iEndTime** As Long
 - o **iProject** As CProject
 - o **iStartTime** As Long
- Membres públics:
 - o Funcions i procediments:
 - **Addtrack** (): afegeix una pista (element CTrack) a la col·lecció de pistes.
 - **Clear** (): elimina totes les pistes de la col·lecció.
 - **RemoveTrack** (ByVal I As Integer): elimina la pista amb índex "I".
 - **SaveTemp** (): crea un arxiu temporal amb l'estat actual del projecte.
 - o Propietats:
 - Get **count** () As Long: número d'elements que hi ha a la col·lecció.
 - Get/Let **EndTime** () As Long: temps final de selecció de les pistes en milisegons.
 - Get **Item** (index As Integer) As CTrack: retorna l'element amb índex "index" de la col·lecció de pistes.
 - Get **MaxLength** () As Long: durada màxima del projecte en milisegons.
 - Get **NewEnum** () As IUnknown
 - Get **Project** () As CProject: retorna l'element de tipus CProject al qual pertany la col·lecció de pistes.

- Get/Let **StartTime** () As Long: temps inicial de selecció de les pistes en milisegons.
- Esdeveniments:
 - **AddedCut** (NewCut As CCut): s'ha afegit un tall.
 - **AddedOrMovedControlPoint** (ControlPoint As CControlPoint): s'ha afegit o modificat un punt de control.
 - **AddedTrack** (NewTrack As CTrack): s'ha afegit una pista.
 - **ChangedCut** (Cut As CCut): s'ha mogut un tall.
 - **ChangedMaxLength** (NewLength As Long): la durada del projecte ha variat.
 - **ChangedStartTime** (NewStart As Long): s'ha modificat el temps inicial de selecció.
 - **FinalCutSelection** (Cut As CCut): ha canviat l'estat de selecció d'un tall.
 - **NewCutInterval** (Cut As CCut): s'ha modificat l'interval reproduïble d'un tall.
 - **RemovedControlPoint** (index As Integer): s'ha eliminat un punt de control.
 - **RemovedCut** (index As Integer): s'ha eliminat un tall.
 - **RemovedTrack** (index As Integer): s'ha eliminat una pista.
 - **SaveTemp** (): s'ha de guardar un arxiu temporal amb l'estat actual del projecte.
- Membres privats:
 - Funcions i procediments:
 - **Class_Terminate** ()
- Membres "Friend":
 - Funcions i procediments:
 - **AddedCut** (NewCut As CCut): s'ha afegit un tall.
 - **ChangedCut** (Cut As CCut): s'ha mogut un tall.
 - **NewCutInterval** (Cut As CCut): s'ha modificat l'interval reproduïble d'un tall.

- **NewOrMovedControlPoint** (ControlPoint As CControlPoint): s'ha afegit o modificat un punt de control.
 - **RemovedControlPoint** (index As Integer): s'ha eliminat un punt de control.
 - **RemovedCut** (index As Integer): s'ha eliminat un tall.
 - **SelectedCut** (Cut As CCut): s'ha seleccionat un tall.
 - **TrackChangedMaxLength** (NewLength As Long): la durada del projecte ha variat
- Propietats:
- Set **Project** (ByRef NewProject As CProject): fixa l'element de tipus CProject al qual pertany la col·lecció de pistes.

CTrack.cls

Aquesta classe és la que representa una pista d'àudio del DSI Visual Track, i és el tipus d'objecte que conforma la col·lecció de pistes que prèviament hem vist a la classe CTracks.

- Atributs privats:
 - o Const **ClassName** As String = "CTrack"
 - o WithEvents **iCuts** As CCuts
 - o **iEndSelTrack** As Long
 - o **iMute** As Boolean
 - o **iSelected** As Boolean
 - o **iSolo** As Boolean
 - o **iStartSelTrack** As Long
 - o **iTracks** As CTracks
- Membres públics:
 - o Funcions i procediments:
 - **DeselectBrothersAndMe** (): estableix totes les pistes com a inactives.
 - o Propietats:
 - Get **Cuts** () As CCuts: retorna la col·lecció de talls de tipus CCuts.
 - Get/Let **EndSelTrack** () As Long: no utilitzat.
 - Get **index** () As Integer: retorna l'índex de la pista amb la que estiguem treballant.
 - Get **MaxLength** () As Long: durada màxima del projecte en milisegons.
 - Get/Let **Mute** () As Boolean: conté l'estat de *mute* d'aquella pista.
 - Get/Let **Selected** () As Boolean: pista activa (*true*) o inactiva (*false*).
 - Get/Let **Solo** () As Boolean: conté l'estat de *solo* d'aquella pista.
 - Get/Let **StartSelTrack** () As Long: no utilitzat.

- Esdeveniments:
 - **ChangedMaxLength** (NewMaxLength As Long): la durada del projecte ha variat.
 - **ChangedMute** (NewMute As Boolean): l'estat de *mute* ha canviat.
 - **ChangedSelection** (NewState As Boolean): l'estat de pista activa a canviat..
 - **ChangedSolo** (NewSolo As Boolean): l'estat de *solo* ha canviat.
- Membres privats:
 - Funcions i procediments:
 - **Class_Initialize** ()
 - **iCuts_AddedCut** (Cut As CCut): s'ha afegit un tall.
 - **iCuts_AddedOrMovedControlPoint** (ControlPoint As CControlPoint): s'ha afegit o modificat un punt de control.
 - **iCuts_CallSaveTemp** (): s'ha de crear un arxiu temporal.
 - **iCuts_ChangedPosition** (Cut As CCut): s'ha mogut un tall.
 - **iCuts_NewInterval** (Cut As CCut): s'ha canviat l'interval visible d'un tall.
 - **iCuts_RemovedControlPoint** (index As Integer): s'ha eliminat un punt de control.
 - **iCuts_RemovedCut** (ByVal index As Integer): s'ha eliminat un tall.
 - **iCuts_SelectedCut** (Cut As CCut): s'ha seleccionat un tall.
- Membres "Friend":
 - Propietats:
 - **Set Tracks** (NewTracks As CTracks): estableix la col·lecció de pistes a la qual pertany la pista amb la que estem treballant.
 - **Get Tracks** () As CTracks: retorna la col·lecció de pistes a la qual pertany la pista amb la que estem treballant.

CCuts.cls

Aquesta classe conté tot el que fa referència als talls en general. La seva funció principal és contenir una col·lecció d'elements tall (CCut).

- Atributs privats:
 - o Const **ClassName** As String = "CCuts"
 - o **iCollection** As New Collection: col·lecció d'elements CCut (talls).
- Membres públics:
 - o Funcions i procediments:
 - **AddCut** (ByVal Name As String, ByVal StartTime As Long) As Boolean: afegeix un tall a la posició indicada.
 - **RemoveCut** (ByVal index As Integer): elimina un tall.
 - **SaveTemp** (): crea un arxiu temporal amb l'estat actual del projecte.
 - **SelectCut** (ByRef Cut As CCut): no utilitzat.
 - **SelectInterval** (ByRef Cut As CCut, IFStartInt As Long, IFEndInt As Long): selecciona l'interval reproduïble d'un tall.
 - **ValidatePosition** (CutToBeExplored As CCut) As Boolean: comprova si un tall es pot afegir.
 - o Propietats:
 - Get **count** () As Long: número d'elements que hi ha a la col·lecció.
 - Get **Item** (index As Integer) As CCut (index As Integer) As CCut: retorna l'element amb índex "index" de la col·lecció de talls.
 - Get **NewEnum** () As IUnknown
 - o Esdeveniments:
 - **AddedCut** (ByRef Cut As CCut): s'ha afegit un tall.
 - **AddedOrMovedControlPoint** (ControlPoint As CControlPoint): s'ha afegit o modificat un punt de control.
 - **CallSaveTemp** (): s'ha de guardar un arxiu temporal amb l'estat actual del projecte.

- **ChangedPosition** (ByRef Cut As CCut): s'ha mogut un tall.
 - **ChangedZoom** (): el zoom ha variat.
 - **NewInterval** (ByRef Cut As CCut): s'ha modificat l'interval reproduïble d'un tall.
 - **RemovedControlPoint** (index As Integer): s'ha eliminat un punt de control.
 - **RemovedCut** (ByVal index As Integer): s'ha eliminat un tall.
 - **SelectedCut** (ByRef Cut As CCut): no utilitzat.
- Membres privats:
- Funcions i procediments:
 - **AreCompatible** (A As CCut, B As CCut) As Boolean: comprova si dos talls poden coexistir a la mateixa pista.
 - **Class_Terminate** ()
- Membres "Friend":
- Funcions i procediments:
 - **NewOrMovedControlPoint** (ControlPoint As CControlPoint): s'ha afegit o modificat un punt de control.
 - **NotifyChangedPosition** (Cut As CCut): s'ha mogut un tall.
 - **RemovedControlPoint** (index As Integer): s'ha eliminat un punt de control.

CCut.cls

Aquesta classe representa un tall de so que està contingut dins de la col·lecció de talls de so que hem creat a la classe anterior CCuts.

- Atributs privats:
 - o Const **ClassName** As String = "CCut"
 - o WithEvents **iControlPoints** As CControlPoints
 - o **iCutName** As String
 - o **iCuts** As CCuts
 - o **iDSAudio** As IBasicAudio
 - o **iDSControl** As IMediaControl
 - o **iDSEvent** As IMediaEvent
 - o **iDSPosition** As IMediaPosition
 - o **iEndInterval** As Long
 - o **iFile** As Integer
 - o **iSelected** As Boolean
 - o **iStartInterval** As Long
 - o **iStartTime** As Long
 - o **RIFF** As New CRIFF

- Membres públics:
 - o Funcions i procediments:
 - **LetCutName** (Name As String) As Boolean: indica si ha pogut carregar el tall de so per a la seva posterior reproducció.
 - **StartGettingEnergy** () As Boolean: indica si serà possible obrir l'arxiu de so i entendre'n el seu format per tal de llegir-ne la informació relativa a la forma d'ona.
 - **StopGettingEnergy** (): tanca l'arxiu de so un cop s'ha acabat la seva lectura.
 - **DeselectBrothersAndMe** (): estableix tots els talls de la col·lecció com a no seleccionats.
 - **PlayCut** ()
 - **PlayFile** () As Boolean: reproduueix l'interval seleccionat del tall amb el que estem treballant.

- **StopCut ()**: atura la reproducció del tall amb el que estem treballant.
- Propietats:
 - **ControlPoints ()** As CControlPoints: retorna la col·lecció de punts de control de tipus CControlPoints.
 - **Get CutName ()** As String: nom del tall de so.
 - **Get/Let EndInterval ()** As Long: temps final de l'interval reproduïble del tall en milisegons.
 - **Get EndTime ()** As Long: posició final del tall a la pista en milisegons.
 - **Get FileLength ()** As Single: durada total del tall de so.
 - **Get GetEnergy (ms As Long) As Single**: valor en dB d'un instant de temps "ms" del tall de so.
 - **Get index ()** As Integer: retorna l'índex del tall amb el que estiguem treballant.
 - **Get IsRIFFFile ()** As Boolean: propietat que ens informa sobre si el format de l'arxiu és RIFF.
 - **Get Length ()** As Single: durada del tall de so tenint en compte l'interval reproduïble.
 - **Get/Let Selected ()** As Boolean: conté l'estat de selecció d'un tall.
 - **Get/Let StartInterval ()** As Long: temps inicial de l'interval reproduïble del tall en milisegons.
 - **Get/Let StartTime ()** As Long: posició inicial del tall a la pista en milisegons.
- Esdeveniments:
 - **ChangedPosition ()**: s'ha mogut el tall.
 - **ChangedSelection (NewState As Boolean)**: s'ha modificat l'estat de selecció del tall.
- Membres privats:
 - Funcions i procediments:
 - **Class_Initialize ()**

- **Class_Terminate** ()
 - **DxTerminate** () As Boolean: descarrega el tall de so dels objectes de DirectX utilitzats per a la seva reproducció.
 - **iControlPoints_AddedControlPoint** (ControlPoint As CControlPoint): s'ha afegit un punt de control.
 - **iControlPoints_ModifiedControlPoint** (ControlPoint As CControlPoint): s'ha modificat un punt de control.
 - **iControlPoints_RemovedControlPoint** (ByVal index As Integer): s'ha eliminat un punt de control.
 - **iControlPoints_ValidateControlPoint** (ControlPoint As CControlPoint): comprova si la posició d'un punt de control és possible tenint en compte els propis límits que imposa la durada del tall.
 - **LoadFileIntoDX** (FileName As String) As Boolean: carrega el tall de so als objectes de DirectX utilitzats per a la seva reproducció.
- Membres "Friend":
- o Propietats:
 - Set **Cuts** (NewCuts As CCuts): estableix la col·lecció de talls a la qual pertany el tall amb el que estem treballant.
 - Get **Cuts** () As CCuts: retorna la col·lecció de talls a la qual pertany el tall amb el que estem treballant.

CControlpoints.cls

Aquesta classe engloba totes les funcions aplicables a un conjunt de punts de control. La seva principal característica és que es comporta com una col·lecció d'elements controlpoint (CControlPoint).

- Atributs privats:
 - o Const **ClassName** As String = "CControlPoints"
 - o **iCollection** As New Collection: col·lecció d'elements CControlPoint (punts de control)
 - o Const **MaxLevel** As Single= 0
 - o Const **MinLevel** As Single= -40
- Membres públics:
 - o Funcions i procediments:
 - **AddControlPoint** (ByVal Position As Long, ByVal Level As Single) As Boolean: intenta afegir un punt de control a la posició i nivell indicats.
 - **ModifyControlPoint** (ByRef MovedControlPoint As CControlPoint, ByVal Position As Long, ByVal Level As Single) As Boolean: intenta modificar la posició i/o nivell d'un punt de control.
 - **RemoveControlPoint** (ByVal index As Integer): elimina un punt de control.
 - o Propietats:
 - Get **Count** () As Long: número d'elements que hi ha a la col·lecció.
 - Get **Item** (index As Integer) As CControlPoint: retorna l'element amb índex "index" de la col·lecció de punts de control.
 - Get **NewEnum** () As IUnknown
 - o Esdeveniments:
 - **AddedControlPoint** (ByRef ControlPoint As CControlPoint): s'ha afegit un punt de control.
 - **ModifiedControlPoint** (ByRef ControlPoint As CControlPoint): s'ha modificat un punt de control.

- **RemovedControlPoint** (ByVal index As Integer): s'ha eliminat un punt de control.
 - **ValidateControlPoint** (ByRef ControlPoint As CControlPoint): pregunta si la posició del punt de control "ControlPoint" és vàlida respecte la durada del tall al qual pertany.
- Membres privats:
- Funcions i procediments:
 - **Class_Terminate** ()

CControlpoint.cls

Aquesta classe representa un punt de control que forma part de la col·lecció de punts de control que hem creat a la classe anterior CControlPoints.

- Atributs privats:
 - o Const **ClassName** As String = "CControlPoint"
 - o **iControlPoints** As CcontrolPoints
 - o **iLevel** As Single
 - o **iPosition** As Long

- Membres públics:
 - o Propietats:
 - Get **index** () As Integer: retorna l'índex del punt de control amb el que estiguem treballant.
 - Get/Let **Level** () As Single: nivell de volum que representa el punt de control en dB.
 - Get/Let **Position** () As Long: posició del punt de control en milisegons.

- Membres "Friend":
 - o Propietats:
 - Set **ControlPoints** (NewControlPoints As CControlPoints): estableix la col·lecció de punts a la qual pertany el punt de control amb el que estem treballant.
 - Get **ControlPoints** () As CControlPoints: retorna la col·lecció de punts a la qual pertany el punt amb el que estem treballant.

2. Descripció dels controls de la interfície

En els següents paràgrafs descriurem el funcionament intern de la interfície, els controls d'usuari, els formularis i les funcions més rellevants que aquests contenen.

IFProject.ctl

Aquest control representa tot un projecte del DSI Visual Track, ja que n'és el component més extern i és el que es relaciona amb la classe CProject del nucli. Dins d'aquest control ja hi trobem el següent, IFTracks.

- Atributs privats:
 - o **InitialNumTracks** As Integer
 - o WithEvents **P** As CProject
 - o WithEvents **T** As CTracks
- Membres públics:
 - o Funcions i procediments:
 - **AddOneTrack** (): afegeix una pista al nucli sense demanar la creació d'un arxiu temporal amb l'estat del projecte.
 - **AllButtonsVisible** (Visible As Boolean): mostra/amaga tots els botons del menú superior.
 - **ButtonVisible** (sKey As String, Visible As Boolean): permet mostrar/amagar un botó en concret del menú superior.
 - **HideLibrary** (): amaga el navegador d'arxius.
 - **HideOrShowLibrary** (): mostra/amaga el navegador d'arxius.
 - **SaveProject** (): guarda l'estat del projecte mitjançant una crida al nucli.
 - o Propietats:
 - Get **Edited** () As Boolean: demana al nucli si un projecte ha estat modificat després d'haver estat guardat.
 - Get/Let **ForeColor** () As Long: color amb el que es visualitzarà la forma d'ona dels talls.
 - Get/Let **Highlight** () As Boolean: informa sobre si hi ha un interval seleccionat que pot ser emès per una emissora de ràdio.

- Get/Let **HighlightEnd** () As Long: temps final de l'interval del projecte que pot ser emès per una emissora de ràdio.
 - Get/Let **HighlightStart** () As Long: temps inicial de l'interval del projecte que pot ser emès per una emissora de ràdio.
 - Get/Let **InitialNumTracks** () As Integer: número de pistes amb les quals s'inicialitzarà el nucli.
- Membres privats:
- o Funcions i procediments:
 - **AddTrack** (): afegeix una pista al nucli i demana la creació d'un arxiu temporal amb l'estat del projecte.
 - **cToolbar1_ButtonClick** (ByVal IButton As Long): es prem un botó del menú superior.
 - **cToolbar1_DropDownPress** (ByVal IButton As Long): es prem un botó d'un dels menús desplegable del menú superior.
 - **Dir1_Change** (): s'utilitza el navegador d'unitats de disc.
 - **Drive1_Change** (): s'utilitza el navegador de directoris.
 - **OpenProject** (): obre un projecte prèviament desat mitjançant una crida al nucli.
 - **P_EnableRedoButton** (State As Boolean): s'ha d'activar/desactivar el botó de refer una acció.
 - **P_EnableUndoButton** (State As Boolean): s'ha d'activar/desactivar el botó de desfer una acció.
 - **Picture1_Resize** ()
 - **Playmode1_Click** (): activa el mode 1 de reproducció a partir de la crida adequada a la funció de reproducció del nucli.
 - **Playmode2_Click** (): mode 2 de reproducció
 - **Playmode3_Click** (): mode 3 de reproducció
 - **Playmode4_Click** (): mode 4 de reproducció
 - **Playmode5_Click** (): mode 5 de reproducció
 - **Playmode6_Click** (): mode 6 de reproducció

- **Playmode7_Click ()**: mode 7 de reproducció
- **Playmode8_Click ()**: mode 8 de reproducció
- **PlayTimeline_Click ()**: reproducció completa o, en cas que n'hi hagi, de la zona seleccionada del projecte.
- **RemoveCut ()**: elimina el tall seleccionat de la pista activa.
- **RemoveTrack ()**: elimina l'última pista del projecte.
- **ResetStartTime_Click ()**: retorna la barra d'inici de reproducció al principi del projecte.
- **StopTimeline_Click ()**: atura qualsevol reproducció del projecte.
- **T_ChangedStartTime (NewStart As Long)**: la posició de la barra d'inici de reproducció ha estat modificada.
- **Timer1_Timer ()**
- **UserControl_Initialize ()**
- **UserControl_InitProperties ()**: inicialitza diverses propietat del control.
- **UserControl_KeyDown (KeyCode As Integer, Shift As Integer)**: es prem alguna tecla.
- **UserControl_ReadProperties (PropBag As PropertyBag)**: llegeix diverses propietat que se li assignen externament al control des del formulari.
- **UserControl_Resize ()**
- **UserControl_WriteProperties (PropBag As PropertyBag)**: escriu les propietat que se li assignen externament al control des del formulari.
- **zoomin_Click ()**: es prem el botó de "Zoom In".
- **zoomout_Click ()**: es prem el botó de "Zoom Out".

IFTracks.ctl

Aquest control representa l'espai on s'allotgen les pistes d'àudio i conté l'"array" de controls IFTrack.

- Atributs privats:
 - o Const **ClassName** As String = "IFTracks"
 - o **iDiwStart** As Long
 - o **iDiwStop** As Long
 - o **InitialLength** As Long= 30000
 - o **iTracks** As CTracks
 - o **ScrollerMaxPosition** As Long

- Membres públics:
 - o Funcions i procediments:
 - **EnableTimer** (): activa el rellotge que conté aquest control.
 - **SetDiwStartStop** (NewDiwStart As Long, NewDiwStop As Long): estableix la finestra de temps visible en pantalla.
 - **Zoom** (Factor As Single): fa variar la finestra de temps visible en pantalla, ja sigui ampliant-la o reduint-la segons el tipus de zoom.

 - o Propietats:
 - Get/Let **DiwStart** () As Long: temps inicial de la finestra visible en pantalla en milisegons.
 - Get/Let **DiwStop** () As Long: temps final de la finestra visible en pantalla en milisegons.
 - Get/Let **ForeColor** () As Long: color amb el que es visualitzarà la forma d'ona dels talls.
 - Get/Let **Highlight** () As Boolean: informa sobre si hi ha un interval seleccionat que pot ser emès per una emissora de ràdio.
 - Get/Let **HighlightEnd** () As Long: temps final de l'interval del projecte que pot ser emès per una emissora de ràdio.
 - Get/Let **HighlightStart** () As Long: temps inicial de l'interval del projecte que pot ser emès per una emissora de ràdio.

- **Get TimeScale ()** As Single: relació entre milisegons i unitats de pantalla.
 - **Set Tracks (NewTracks As CTracks):** estableix la classe del nucli a la qual fa referència aquest control.
- Membres privats:
- Funcions i procediments:
 - **CalculateScrollerFromMs (Position As Long) As Integer:** calcula la posició de la barra de desplaçament a partir d'un temps donat.
 - **HScroll1_Change ():** procediment hi ha un canvi a la barra de desplaçament.
 - **HScroll1_Scroll ():** la barra de desplaçament es mou.
 - **IFTrack1_LineMouseDown (Index As Integer, X As Long):** s'ha de canviar la posició de la barra d'inici.
 - **IFTrack1_RequestDiwStartStop (Index As Integer, NewStart As Long, NewStop As Long):** s'ha de canviar la finestra visible en pantalla.
 - **IFTrack1_SelectingMove (Index As Integer, X As Long):** s'ha de fer una selecció d'un tram del projecte.
 - **iTracks_AddedTrack (NewTrack As CTrack):** el nucli ha afegit una pista.
 - **iTracks_ChangedMaxLength (NewLength As Long):** la durada màxima del projecte ha canviat.
 - **iTracks_ChangedStartTime (NewStart As Long):** el temps inicial de reproducció ha canviat.
 - **iTracks_RemovedTrack (Index As Integer):** el nucli ha eliminat una pista.
 - **MsToDisplayUnits (PositionInMs As Long) As Single:** calcula quina posició en pantalla es correspon amb un temps determinat.
 - **PaintHighlight ():** actualitza el pintat de la zona "Highlight".
 - **Picture1_Resize ()**
 - **RecalculateScrollBar ():** actualitza la posició de la barra de desplaçament.

- Propietats:
 - **Get Project ()** As CProject: retorna el projecte al qual pertany l'element de tipus CTracks que representa aquest control.

IFTrack.ctl

Aquest control representa independentment una pista de les quals es compon IFTracks. Per poder disposar de múltiples pistes, hem definit aquest control com un “array” (sèrie), on l’element generador té índex zero i és invisible.

- Atributs privats:
 - o **ClassName** As String = "IFTrack"
 - o **iCuts** As CCuts
 - o **iTrack** As CTrack

- Membres públics:
 - o Funcions i procediments:
 - **PaintHighlight** (): actualitza el pintat de la zona “Highlight”.
 - **SetDiwStartStop** (NewDiwStart As Long, NewDiwStop As Long): estableix la finestra de temps visible en pantalla.

 - o Propietats:
 - Get/Let **DiwStart** () As Long: temps inicial de la finestra visible.
 - Get/Let **DiwStop** () As Long: temps final de la finestra visible.
 - Get/Let **ForeColor** () As Long: color amb el que es visualitzarà la forma d’ona dels talls.
 - Get/Let **Highlight** () As Boolean: informa sobre si hi ha un interval seleccionat que pot ser emès per una emissora de ràdio.
 - Get/Let **HighlightEnd** () As Long: temps final de l’interval del projecte que pot ser emès per una emissora de ràdio.
 - Get/Let **HighlightStart** () As Long: temps inicial de l’interval del projecte que pot ser emès per una emissora de ràdio.
 - Get/Let **Selected** () As Boolean: informa si aquesta pista és activa o inactiva.
 - **Set Track** (NewTrack As CTrack): estableix la classe del nucli a la qual fa referència aquest control.

 - o Esdeveniments:
 - **LineMouseDown** (X As Long): s’ha de canviar la posició de la barra d’inici.

- **RequestDiwStartStop** (NewStart As Long, NewStop As Long): s'ha de canviar la finestra visible en pantalla.
- **SelectingMove** (X As Long): s'ha de canviar l'interval seleccionat del projecte.
- Membres privats:
 - o Funcions i procediments:
 - **IFCuts1_GotFocus** (): estem treballant amb aquesta pista en concret.
 - **IFCuts1_LineMouseDown** (X As Long): s'ha de canviar la posició de la barra d'inici.
 - **IFCuts1_RequestDiwStartStop** (NewStart As Long, NewStop As Long): s'ha de canviar la finestra visible en pantalla.
 - **IFCuts1_SelectingMove** (X As Long): s'ha de canviar el tram seleccionat del projecte.
 - **iTrack_ChangedMute** (NewMute As Boolean): procediment el nucli ha canviat l'estat de *mute* d'aquesta pista.
 - **iTrack_ChangedSelection** (NewState As Boolean): el nucli ha canviat l'estat d'activa/inactiva d'aquesta pista.
 - **iTrack_ChangedSolo** (NewSolo As Boolean): el nucli ha canviat l'estat de *solo* d'aquesta pista.
 - **mute_Click** (): premem el botó de *solo*.
 - **ShowSelectedColor** (NewState As Boolean): actualitza el pintat de pista activa/inactiva d'aquesta pista.
 - **solo_Click** (): premem el botó de *mute*.
 - **UserControl_Resize** ()

IFCuts.cls

Aquest control s'encarrega de representar gràficament l'espai on s'afegiran els talls dins de cada pista i conté l'"array" de control IFCut.

- Atributs privats:
 - o Const **BorderWidth** = 30
 - o Const **ClassName** As String= "IFCuts"
 - o **Dragging** [As Boolean]
 - o **DraggingInitialExX** As Single
 - o **DraggingInitialInX** As Single
 - o WithEvents **iCuts** As CCuts
 - o **iDiwStart** As Long
 - o **iDiwStop** As Long
 - o **iHighlight** As Boolean
 - o **iHighlightStart** As Long
 - o **InitialInterval** As Single
 - o **InitialTime** As Single
 - o **Initial** As Single
 - o **IntervalingStart** As Boolean
 - o WithEvents **iTracks** As CTracks
 - o **Selecting** As Boolean

- Membres públics:
 - o Funcions i procediments:
 - **PaintHighlight** (): actualitza el pintat de la zona "Highlight".
 - **Refresh** (): treu tots els talls que hi hagi en una pista i els torna a carregar.
 - **Reposition** (): actualitza la posició dels talls d'una pista.
 - **RepositionCut** (C As CCut): actualitza la posició d'un tall d'una pista.
 - **SetDiwStartStop** (NewDiwStart As Long, NewDiwStop As Long): estableix la finestra de temps visible en pantalla.

- Propietats:
 - **Set Cuts** (NewCuts As CCuts): estableix la classe del nucli a la qual fa referència aquest control.
 - **Get/Let DiwStart** () As Long: temps inicial de la finestra visible.
 - **Get/Let DiwStop** () As Long: temps final de la finestra visible.
 - **Get/Let ForeColor** () As Long: color amb el que es visualitzarà la forma d'ona dels talls.
 - **Get/Let Highlight** () As Boolean: informa sobre si hi ha un interval seleccionat que pot ser emès per una emissora de ràdio.
 - **Get/Let HighlightEnd** () As Long: temps final de l'interval del projecte que pot ser emès per una emissora de ràdio.
 - **Get/Let HighlightStart** () As Long: temps inicial de l'interval del projecte que pot ser emès per una emissora de ràdio.
- Esdeveniments:
 - **LineMouseDown** (X As Long): s'ha de canviar la posició de la barra d'inici.
 - **RequestDiwStartStop** (NewStart As Long, NewStop As Long): s'ha de canviar la finestra visible en pantalla.
 - **SelectingMove** (X As Long): s'ha de canviar l'interval seleccionat del projecte.
- Membres privats:
 - Funcions i procediments:
 - **CreateIFCut** (C As CCut): afegeix un tall a l'"array" de talls IFCut.
 - **iCuts_AddedCut** (NewCut As CCut): el nucli ha afegit un tall.
 - **iCuts_ChangedPosition** (Cut As CCut): el nucli ha mogut un tall.
 - **iCuts_NewInterval** (Cut As CCut): procediment el nucli ha modificat l'interval visible d'un tall.
 - **iCuts_RemovedCut** (ByVal Index As Integer): el nucli ha eliminat un tall.

- **IFCut1_LineMouseDown** (Index As Integer, X As Long): s'ha de canviar la posició de la barra d'inici.
- **IFCut1_MouseDown** (Index As Integer, X As Single): procediment que s'executa quan s'ha de començar a moure un tall.
- **IFCut1_MouseDownEndInterval** (Index As Integer, X As Single): s'ha de començar a modificar l'interval visible d'un tall per l'extrem final.
- **IFCut1_MouseDownStartInterval** (Index As Integer, X As Single): s'ha de començar a modificar l'interval visible d'un tall per l'extrem inicial.
- **IFCut1_MouseMove** (Index As Integer, X As Single): es vol moure un tall.
- **IFCut1_MouseMoveInterval** (Index As Integer, X As Single): es vol modificar l'interval visible.
- **IFCut1_MouseUp** (Index As Integer, X As Single): es vol deixar de moure un tall.
- **IFCut1_MouseUpInterval** (Index As Integer, X As Single): es vol deixar de modificar l'interval visible d'un tall.
- **IFCut1_SelectingMove** (Index As Integer, X As Long): s'ha de canviar el tram seleccionat del projecte.
- **MsToDisplayUnits** (PositionInMs As Long) As Single: calcula quina posició en pantalla es correspon amb un temps determinat.
- **ScrollIfNecessary** (DesiredPosition As Long): comprova si cal actualitzar la posició de la barra de desplaçament per mostrar un instant de temps en concret.
- **UserControl_MouseDown** (Button As Integer, Shift As Integer, X As Single, Y As Single): proc es prem el ratolí sobre el control.
- **UserControl_MouseMove** (Button As Integer, Shift As Integer, X As Single, Y As Single): es desplaça el ratolí per damunt del control.
- **UserControl_MouseUp** (Button As Integer, Shift As Integer, X As Single, Y As Single): es deixa de prémer el ratolí sobre el control.

- **UserControl_OLEDragDrop** (Data As DataObject, Effect As Long, Button As Integer, Shift As Integer, X As Single, Y As Single): es deixa anar un arxiu sobre el control.
- **UserControl_Resize** ()
- Propietats:
 - **Get TimeScale** () As Single: retorna la relació entre milisegons i unitats de pantalla.
 - **Set Tracks** (NewTracks As CTracks): no utilitzat.

IFCut.ctl

Aquest control és la representació gràfica d'un tall de so que podrà ser inserit a IFCut. Per poder treballar amb múltiples talls, hem definit aquest control com un "array" (sèrie), on l'element generador té índex zero i és invisible.

- Atributs privats:
 - o **BorderWidth** = 30
 - o **ClassName** As String = "IFCut"
 - o **iControlPoints** As CControlPoints
 - o **iCut** As CCut
 - o **iForeColor** As Long
- Membres públics:
 - o Propietats:
 - Set **Cut** (NewCut As CCut): estableix la classe del nucli a la qual fa referència aquest control.
 - Get/Let **ForeColor** () As Long: color amb el que es visualitzarà la forma d'ona dels talls.
 - Get/Let **Selected** () As Boolean: informa sobre si el tall està seleccionat o no.
 - o Esdeveniments:
 - **CutGotFocus** (): no utilitzat.
 - **DesireToDrag** (): no utilitzat.
 - **LineMouseDown** (X As Long): s'ha de canviar la posició de la barra d'inici.
 - **MouseDown** (X As Single): volem començar a moure un tall.
 - **MouseDownEndInterval** (X As Single): volem començar a modificar l'interval visible d'un tall per l'extrem final.
 - **MouseDownStartInterval** (X As Single): volem començar a modificar l'interval visible d'un tall per l'extrem inicial.
 - **MouseMove** (X As Single): volem moure un tall.
 - **MouseMoveInterval** (X As Single): volem modificar l'interval visible d'un tall.

- **MouseUp** (X As Single): volem deixar de moure un tall.
 - **MouseUpInterval** (X As Single): volem deixar de modificar l'interval visible d'un tall.
 - **SelectingMove** (X As Long): esdeveniment s'ha de canviar l'interval seleccionat del projecte.
- Membres privats:
- o Funcions i procediments:
 - **GrabWaveForm** (): pinta la forma d'ona del tall.
 - **iCut_ChangedSelection** (NewState As Boolean): hi ha hagut un canvi en l'estat de selecció del tall.
 - **IFControlPoints1_LineMouseDown** (X As Long): s'ha de canviar la posició de la barra d'inici.
 - **IFControlPoints1_SelectingMove** (X As Long): s'ha de canviar el tram seleccionat del projecte.
 - **Label1_Click** (): es clica damunt de l'etiqueta del tall.
 - **Label1_MouseDown** (Button As Integer, Shift As Integer, X As Single, Y As Single): es prem el ratolí damunt de l'etiqueta.
 - **Label1_MouseMove** (Button As Integer, Shift As Integer, X As Single, Y As Single): es mou el ratolí damunt de l'etiqueta.
 - **Label1_MouseUp** (Button As Integer, Shift As Integer, X As Single, Y As Single): es deixa anar el ratolí damunt de l'etiqueta.
 - **Picture1_Resize** ()
 - **Repaint** (): actualitza el dibuix de la forma d'ona.
 - **ShowMenu** (X As Single, Y As Single): mostra el menú d'opcions del tall.
 - **ShowSelectedColor** (NewState As Boolean): actualitza el dibuix de l'estat de selecció del tall.
 - **UserControl_Click** (): es clica damunt del control.
 - **UserControl_Initialize** ()

- **UserControl_MouseDown** (Button As Integer, Shift As Integer, X As Single, Y As Single): es prem el ratolí sobre el control.
- **UserControl_MouseMove** (Button As Integer, Shift As Integer, X As Single, Y As Single): es mou el ratolí sobre el control.
- **UserControl_MouseUp** (Button As Integer, Shift As Integer, X As Single, Y As Single): es deixa anar el ratolí sobre el control.
- **UserControl_Resize** ()

IFControlPoints.ctl

Aquest control s'encarrega de representar gràficament l'espai on s'afegiran els punts de control de volum dins de cada tall i conté l'"array" de control IFControlPoint.

- Atributs privats:
 - o **ClassName** As String = "IFControlPoints"
 - o **Dragging** As Boolean
 - o **DraggingInitialExX** As Single
 - o **DraggingInitialExY** As Single
 - o **DraggingInitialInX** As Single
 - o **DraggingInitialInY** As Single
 - o **iControlPoints** As CControlPoints
 - o **iCut** As CCut
 - o **iCuts** As CCuts
 - o **iWaveForm** As Picture
 - o **Selecting** As Boolean
- Membres públics:
 - o Funcions i procediments:
 - **GrabWaveForm** (P As Picture): pinta la forma d'ona del tall a partir de la que pinta IFCut.
 - **Refresh** (): treu tots els punts de control que hi hagi en una tall i els torna a carregar.
 - **Reposition** (): actualitza la posició dels punts de control d'un tall.
 - o Propietats:
 - Set **Cut** (NewCut As CCut): estableix el tall del nucli al qual pertany aquesta col·lecció de punts de control.
 - Set **Cuts** (NewCuts As CCuts): no utilitzat.
 - o Esdeveniments:
 - **LineMouseDown** (X As Long): esdeveniment que indica que s'ha de canviar la posició de la barra d'inici.

- **SelectingMove** (X As Long): s'ha de canviar l'interval seleccionat del projecte.
- Membres privats:
 - o Funcions i procediments:
 - **CreateIFControlPoint** (C As CControlPoint): afegeix un punt de control a l'"array" de punts IFControlPoints.
 - **iControlPoints_AddedControlPoint** (ControlPoint As CControlPoint): s'ha afegit un punt de control al nucli.
 - **iControlPoints_ModifiedControlPoint** (ControlPoint As CControlPoint): s'ha modificat un punt de control al nucli.
 - **IFControlPoint1_DoubleClick** (Index As Integer): es fa doble clic al damunt d'un punt de control.
 - **IFControlPoint1_MouseDown** (Index As Integer, X As Single, Y As Single): es prem el ratolí al damunt d'un punt de control.
 - **IFControlPoint1_MouseMove** (Index As Integer, X As Single, Y As Single): es mou el ratolí al damunt d'un punt de control.
 - **IFControlPoint1_MouseUp** (Index As Integer, X As Single, Y As Single): es deixa anar el ratolí al damunt d'un punt de control.
 - **MoveControlPoint** (Index As Integer, X As Single, Y As Single): intenta moure un punt de control al nucli.
 - **PaintLines** (): pinta les línies que uneixen els punts de control.
 - **RepositionControlPoint** (C As CControlPoint): actualitza la posició d'un punt de control.
 - **UserControl_Initialize** ()
 - **UserControl_MouseDown** (Button As Integer, Shift As Integer, X As Single, Y As Single): es prem el ratolí sobre el control.
 - **UserControl_MouseMove** (Button As Integer, Shift As Integer, X As Single, Y As Single): es mou el ratolí sobre el control.
 - **UserControl_MouseUp** (Button As Integer, Shift As Integer, X As Single, Y As Single): es deixa anar el ratolí sobre el control.
 - **UserControl_Resize** ()

- Propietats:

- Set **ControlPoints** (NewControlPoints As CControlPoints): estableix la classe del nucli a la qual fa referència aquest control.
- Get **LevelScale** () As Single: relació entre dB i unitats de pantalla.
- Get **TimeScale** () As Single: relació entre milisegons i unitats de pantalla.

IFControlPoint.ctl

Aquest control representa independentment una punt de control de volum dels quals es compon IFControlPoints. Per poder disposar de múltiples punts, hem definit aquest control com un “array” (sèrie), on l’element generador té índex zero i és invisible.

- Atributs privats:
 - o **ClassName** As String = "IFControlPoint"
 - o **iControlPoint** As CControlPoint
- Membres públics:
 - o Esdeveniments:
 - **DoubleClick** (): s’ha fet doble clic sobre el control per eliminar un punt de control.
 - **MouseDown** (X As Single, Y As Single): s’ha premut el ratolí sobre el control per començar a moure un punt de control.
 - **MouseMove** (X As Single, Y As Single): s’ha mogut el ratolí sobre el control per moure un punt de control.
 - **MouseUp** (X As Single, Y As Single): s’ha deixat de prémer el ratolí sobre el control per deixar de moure un punt de control.
- Membres privats:
 - o Funcions i procediments:
 - **UserControl_DbIClick** (): es fa doble clic sobre el control.
 - **UserControl_MouseDown** (Button As Integer, Shift As Integer, X As Single, Y As Single): es prem el ratolí al damunt del control.
 - **UserControl_MouseMove** (Button As Integer, Shift As Integer, X As Single, Y As Single): es mou el ratolí pel damunt del control.
 - **UserControl_MouseUp** (Button As Integer, Shift As Integer, X As Single, Y As Single): es deixa de prémer el ratolí al damunt del control.
 - o Propietats:
 - Set **ControlPoint** (NewPoint As CControlPoint): estableix la classe del nucli a la qual fa referència aquest control.

3. Objecte AMTimelineObj

URL: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directshow/htm/amtimelineobjobject.asp>

AMTimelineObj Object

Provides methods for manipulating [timeline](#) objects in [DirectShow Editing Services](#) (DES). All timeline objects support **AMTimelineObj**, including source, effect, transition, track, group, and composition objects. Create a timeline object by calling the [AMTimeline.CreateEmptyNode](#) method.

This object has the following methods.

Method	Description
ClearDirty	Not supported for Visual Basic.
FixTimes	Not supported for Visual Basic.
FixTimes2	Rounds the specified start and stop times to the nearest frame boundaries, as defined by the parent group's frame rate setting.
GetDirtyRange	Not supported for Visual Basic.
GetDirtyRange2	Not supported for Visual Basic.
GetEmbedDepth	Not supported for Visual Basic.
GetGenID	Retrieves the object's generated identifier.
GetGroupIBelongTo	Not supported for Visual Basic.
GetLocked	Retrieves the object's editing state (locked or unlocked).
GetMuted	Retrieves the object's muted state.
GetPropertySetter	Not supported for Visual Basic.
GetStartStop	Not supported for Visual Basic.
GetStartStop2	Retrieves the object's start and stop times, relative to the object's parent.
GetSubObject	Not supported for Visual Basic.
GetSubObjectGUID	Not supported for Visual Basic.
GetSubObjectGUIDB	Retrieves the GUID of the subobject associated with this object.
GetSubObjectLoaded	Determines whether a subobject has been created for the object.
GetTimelineNoRef	Not supported for Visual Basic.
GetTimelineType	Retrieves the object's type.
GetUserData	Retrieves application-defined persistent data.
GetUserID	Retrieves the object's application-defined identifier.

GetUserName	Retrieves the object's application-defined name.
Remove	Removes this object from the timeline, for reinsertion elsewhere.
RemoveAll	Permanently removes this object from the timeline, including subobjects and children.
SetDirtyRange	Not supported for Visual Basic.
SetDirtyRange2	Not supported for Visual Basic.
SetLocked	Sets the object's editing state to locked or unlocked.
SetMuted	Sets the object's muted state.
SetPropertySetter	Not supported for Visual Basic.
SetStartStop	Not supported for Visual Basic.
SetStartStop2	Sets the object's start and stop times, relative to the object's parent.
SetSubObject	Not supported for Visual Basic.
SetSubObjectGUID	Not supported for Visual Basic.
SetSubObjectGUIDB	Specifies the GUID of the subobject associated with this object.
SetTimelineType	Not supported for Visual Basic.
SetUserData	Sets application-defined persistent data.
SetUserID	Sets an application-defined identifier for the object.
SetUserName	Sets an application-defined name for the object.