



eetac

Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Desenvolupament d'una aplicació HbbTV per a esdeveniments esportius

TITULACIÓ: Doble titulació en Enginyeria Tècnica Aeronàutica, especialitat Aeronavegació i Enginyeria Tècnica de Telecomunicació, especialitat Sistemes de Telecomunicació

AUTOR: Àlex Ramonjoan Escobar

DIRECTORS: David Rincón Rivera, Alberto J. González Cela

DATA: 19 de juliol de 2012

Títol: Desenvolupament d'una aplicació HbbTV per a esdeveniments esportius

Autor: Àlex Ramonjoan Escobar

Directors: David Rincón Rivera, Alberto J. González Cela

Data: 19 de juliol de 2012

Resum

La televisió connectada s'està erigint com un dels fenòmens tecnològics més destacats dels últims anys en el camp de les telecomunicacions. A nivell europeu cal destacar la importància de l'estàndard *Hybrid Broadcast Broadband TV*, altrament conegut com HbbTV. Es tracta d'una plataforma oberta que té l'objectiu de proporcionar accés a aplicacions interactives i a continguts sota demanda als consumidors, independentment de la marca del terminal o dispositiu de que disposin.

Les perspectives d'èxit d'implantació d'aquest estàndard són relativament altes respecte a anteriors intents degut principalment a que no es tracta d'una nova tecnologia que comporti uns costos elevats de desenvolupament, sinó que HbbTV és una conjunció de l'actual tecnologia de difusió de la televisió digital amb Internet.

En aquest TFC s'ha implementat una aplicació HbbTV específica per a esdeveniments esportius, que permet enriquir l'experiència de l'espectador a partir de l'accés immediat a estadístiques i a les repeticions de les jugades més destacades des de varis punts de vista. El desenvolupament s'ha dut a terme amb l'ajuda d'emuladors de receptor HbbTV. A més, s'ha recreat un escenari realista d'emissions *broadcast* que ens ha permès treballar la totalitat de la cadena de radiodifusió: generació d'un *Transport Stream* amb l'aplicació HbbTV senyalitzada, transmissió del senyal modulat en DVB-T i recepció en terminals compatibles amb l'estàndard (una Smart TV i un *set-top box*).

Finalment s'han fet proves de funcionalitat de l'aplicació amb els diferents dispositius receptors de que hem disposat. S'ha comprovat que el comportament no es el mateix en tots els dispositius, de manera que s'han generat versions adaptades a cada un fins aconseguir-ne una que funciona correctament en tots.

Title: Development of an HbbTV application for sport events

Author: Àlex Ramonjoan Escobar

Directors: David Rincón Rivera, Alberto J. González Cela

Date: July, 19th 2012

Overview

In recent years, connected TV has become one of the most remarkable technologic phenomena in the telecommunications sector around the world. At European level, the relevance of the *Hybrid Broadcast Broadband TV* standard (better known as HbbTV) must be emphasized. The aim of this open platform is to provide access to interactive applications and on-demand contents regardless of the consumer's TV manufacturer.

Prospects for a successful implementation are higher than previous attempts with other standards because of one simple but important reason: it is not a new technology with high development costs. HbbTV is the union of two well known technologies: current digital television broadcasting and Internet.

In this degree thesis an HbbTV application specific for sport events has been designed and implemented. It allows to enrich the TV consumption experience by enabling access to the match statistics and the most interesting moves repetitions from different points of view. The development has been carried out with HbbTV receiver emulators. In addition, a real broadcast emission environment has been recreated for working with the whole broadcasting process: Transport Stream generation with the HbbTV application signaled, DVB-T modulated signal transmission and reception with HbbTV compliant devices (Smart TV and set-top box).

Finally, we have carried out some application functionality tests with our different HbbTV receivers. As it has been demonstrated, the behaviour of the application running on the different devices is not the same, so we have generated some adapted-to-device versions of it. The last version of the HbbTV application works on all the available receivers.

Als excompanys del departament de
DETEC d'Abertis Telecom per
introduir-me en aquest món.

Agraïr l'interès i la dedicació del
David i l'Alberto.

ÍNDIX

INTRODUCCIÓ	1
CAPÍTOL 1. TELEVISIÓ CONNECTADA.....	4
1.1. La xarxa broadcast	4
1.1.1. Xarxes de contribució, distribució i difusió de TV digital	4
1.1.2. DVB: <i>Digital Video Broadcasting</i>	6
1.1.3. Senyal audiovisual. <i>MPEG-2 Systems</i>	7
1.1.4. Senyalització.....	10
1.2. La xarxa broadband	11
1.2.1. Els protocols TCP/IP	12
1.2.2. El protocol HTTP	12
1.3. L'estàndard HbbTV (Hybrid broadcast broadband TV)	13
1.3.1. Introducció	14
1.3.2. Funcionament.....	15
1.3.3. Tipus d'aplicacions	16
1.3.4. Escenaris	17
1.3.5. Tecnologies utilitzades a HbbTV	18
1.3.6. Aplicacions <i>broadcast-related</i>	21
1.3.7. La taula AIT	22
1.3.8. Format dels continguts	22
1.3.9. Protocols de xarxa.....	23
CAPÍTOL 2. DISSENY D'APLICACIONS HBBTV	25
2.1. Definició d'una aplicació DAE.....	25
2.2. Model d'aplicació DAE: cicle de vida	26
2.2.1. Creació d'una aplicació	26
2.2.2. Finalització d'una aplicació.....	27
2.3. DSM-CC Stream Events	28
CAPÍTOL 3. DISSENY I IMPLEMENTACIÓ D'UNA APLICACIÓ HBBTV PER A UN EVENT ESPORTIU	29
3.1. Requisits.....	29
3.2. Arquitectura i entorn de desenvolupament	30
3.2.1. Arquitectura software.....	30
3.2.2. Entorns d'execució durant el desenvolupament.....	31
3.3. Disseny i implementació.....	34
3.3.1. Aplicació HbbTV	34
3.3.2. <i>Webportal</i> d'actualització de continguts	40
CAPÍTOL 4. PROVES I RESULTATS	45

4.1	Generació del Transport Stream.....	45
4.2	Modulació i emissió del Transport Stream	47
4.3	Recepció amb el set-top box EN2000 d'Engel.....	49
4.4	Recepció amb la televisió Samsung SmartTV.....	50
CONCLUSIONS I LÍNIES FUTURES		55
5.1	Conclusions	55
5.2	Línies futures.....	55
BIBLIOGRAFIA		57
GLOSSARI.....		61
ANNEXOS.....		63
Annex A. Diagrames de comportament de les aplicacions		63
Annex B. Anàlisi estructural de fitxers MP4 amb AtomicParsley		67
Annex C. Leaflet de la tarja moduladora DTA-111		69
Annex D. Leaflet del software DTC-300 StreamXpress		70
Annex E. Cody Python d'Opencaster per a generar les taules PSI/SI		71
Annex F. Funcions per a gestionar cookies.....		77
Annex G. Codi de configuració del widget de Twitter		78
Annex H. Captures de les diferents versions de l'aplicació HbbTV desenvolupada		79
	Versió completa amb FireHbbTV	79
	Versió completa amb Opera TV Emulator.....	83
	Versió Engel amb EN2000	87
	Versió Samsung amb Samsung SmartTV.....	91

INTRODUCCIÓ

Al món de la producció audiovisual i la teledifusió ha aparegut recentment el concepte de la TV híbrida, que consisteix en la unió dues tecnologies diferents: la radiodifusió de la televisió digital (*broadcast*) i Internet (*broadband*). El contingut clàssic, televisió lineal (en directe), arriba a l'usuari pel canal unidireccional *broadcast*, mentre que la interactivitat (contingut no lineal) es transmet bidireccionalment per la xarxa IP. D'aquesta manera es pretén donar als consumidors la total llibertat per a que puguin veure qualsevol contingut audiovisual en qualsevol moment utilitzant serveis a la carta. D'altra banda, també es permet incorporar les xarxes socials al nou concepte de consum televisiu, en el qual l'espectador pren un protagonisme inexistent fins el moment.

L'estàndard HbbTV (*Hybrid Broadcast Broadband TV*) [1] s'emmarca dins el context europeu de la televisió híbrida. La primera versió de l'estàndard fou aprovada l'any 2010 per la ETSI. Els principals països impulsors de la norma són França, Alemanya i Espanya. A més, està previst que es facin proves aquest mateix any 2012 fora d'Europa, concretament als Estats Units i al Japó. En l'actualitat la Unió Europea està impulsant la creació de plataformes blanques¹ a través de la EBU per a distribuir-les de manera gratuïta durant esdeveniments internacionals com els Jocs Olímpics de Londres 2012. D'aquesta manera es pretén acostar aquesta nova eina als operadors de radiodifusió, amb l'objectiu d'aconseguir una implantació definitiva de l'estàndard.

En aquest TFC s'ha volgut dissenyar i implementar una aplicació HbbTV amb l'objectiu de fer-ne una eina útil i atractiva per al consumidor de televisió. S'ha optat per desenvolupar una aplicació específica per a esdeveniments esportius, particularitzada en el nostre cas en un partit de futbol. Així doncs, l'aplicació dissenyada permet accedir a les repeticions de les jugades més destacades del partit, així com a les estadístiques i alineacions dels dos equips. S'han intentat implementar altres funcionalitats: reproducció simultània de l'emissió *broadcast* i d'un vídeo servit per la xarxa *broadband*, inclusió de Twitter i gestió de l'actualització de la llista de jugades disponibles (de manera escalable a milions de terminals) mitjançant la senyalització de *Stream Events* a través del flux *broadcast* de DVB-T.

A banda de desenvolupar l'aplicació des del punt de vista de les funcionalitats disponibles per a l'espectador, també s'ha volgut facilitar la tasca de l'operador encarregat de crear i gestionar els continguts audiovisuals interactius. Així doncs, s'ha implementat un *webportal* de gestió de la base de dades on s'emmagatzema tota la informació disponible. L'objectiu és reduir les accions que ha de dur a terme el productor per actualitzar els continguts, de manera que l'interval temporal entre el moment en que es produeix la jugada i l'instant

¹ Les plataformes blanques són aplicacions base no personalitzades que poden ser adaptades a les necessitats de diferents operadors de radiodifusió a partir de canvis de grafisme.

en que les repeticions d'aquesta estan disponibles per a l'espectador sigui mínim i, per tant, atractiu des del punt de vista comercial.

A la Fig i.1 es mostra en un nivell molt abstracte l'escenari complet en el qual seria útil el TFC desenvolupat. L'operador encarregat de la gestió dels continguts de l'aplicació es podria trobar o bé a l'estudi mòbil (furgoneta) o bé als estudis de producció. En tot cas, ell és l'encarregat de pujar els continguts a la xarxa i d'actualitzar la senyalització *broadcast* per a incloure *Stream Events* d'avís a l'espectador. Més endavant es veurà com s'ha recreat l'escenari global per a fer proves no només de les funcionalitats de l'aplicació, sinó també de la part corresponent a l'emissió *broadcast*: generació d'un *Transport Stream* amb l'aplicació HbbTV senyalitzada, modulació, transmissió DVB-T i recepció amb dispositius reals (*set-top box* i Smart TV).

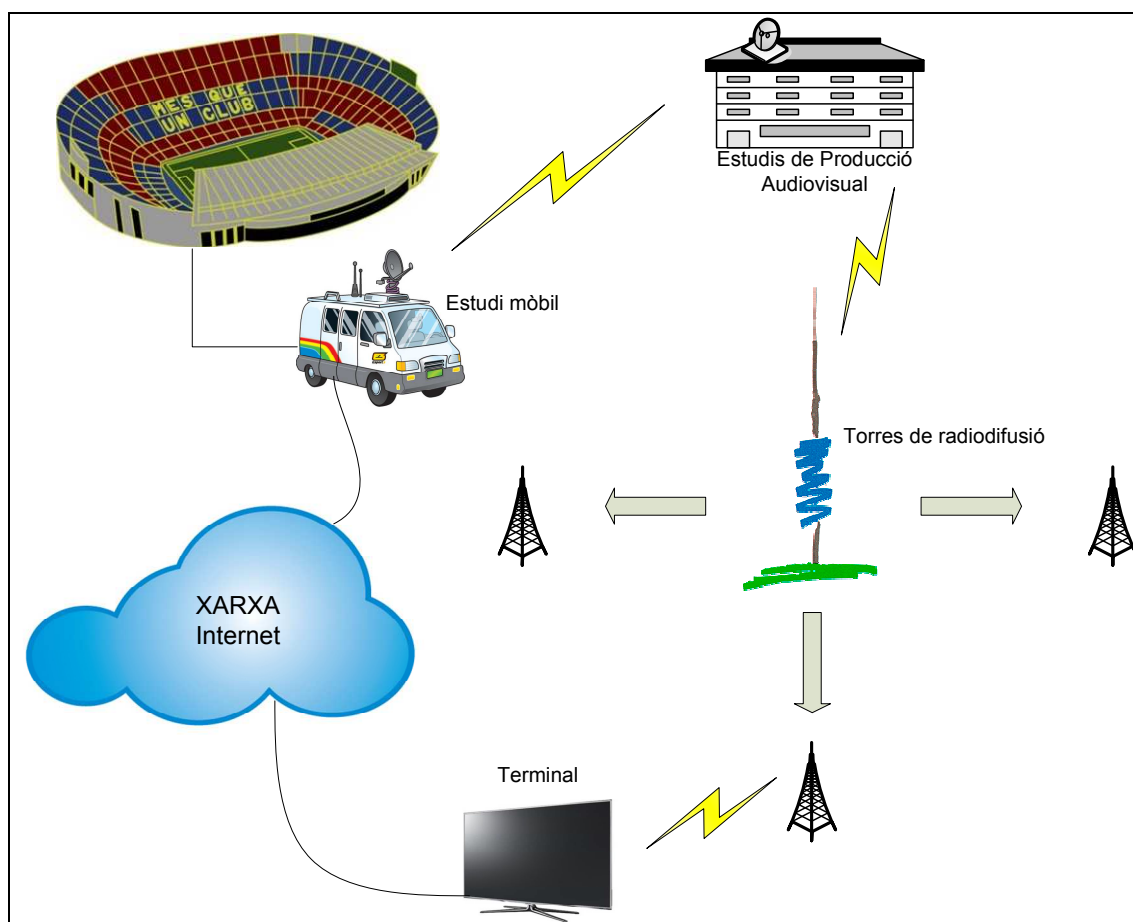


Fig. i.1 Representació abstracta d'un escenari real

Centrant-nos en la memòria, aquesta es divideix en tres capítols. En el primer es fa una introducció a la televisió connectada, explicant les característiques bàsiques de la xarxa *broadcast* i de la xarxa *broadband*. A més, també es fa un resum de la primera versió de l'estàndard HbbTV.

En el segon capítol es fa referència al disseny genèric d'aplicacions HbbTV seguint les pautes marcades per la primera versió del document OIPF *Declaration Environment Application*, tal com assenyala el propi estàndard HbbTV.

Al tercer capítol s'hi exposa el desenvolupament específic de l'aplicació HbbTV dissenyada i el *webportal* d'actualització de continguts. A més, es mostren les eines utilitzades durant el disseny i la implementació.

En el quart capítol es fa un resum de les proves que s'han realitzat al laboratori sobre terminals HbbTV (una Smart TV i un descodificador de TDT o *set-top box*).

Finalment, a les conclusions s'assenyalen els aspectes més rellevants del desenvolupament del TFC i es donen les pautes a seguir de cara a una possible continuació del treball realitzat.

CAPÍTOL 1. TELEVISIÓ CONNECTADA

En aquest primer capítol es dóna la base teòrica per entendre l'entorn de la televisió connectada. D'una banda es parla de la xarxa *broadcast* de televisió digital tal com l'entendem avui en dia; de l'altra, s'introdueixen també els conceptes més essencials de la xarxa *broadband* d'Internet. Finalment es dedica un apartat a la introducció de l'estàndard HbbTV.

1.1. La xarxa *broadcast*

1.1.1. Xarxes de contribució, distribució i difusió de TV digital

La xarxa *broadcast* és la que s'encarrega de fer arribar els continguts generats pels productors audiovisuals i distribuïts pels operadors de radiodifusió als terminals d'una certa regió o zona geogràfica. Podem identificar tres subxarxes dins la xarxa troncal:

- **Xarxa de contribució:** transporta el senyal audiovisual des del seu punt de generació o obtenció fins al centre de producció (CP). El senyal es processa en estudis prèviament a incorporar-se a l'emissió.
- **Xarxa de distribució:** transporta el senyal audiovisual des del centre de producció fins el centre emissor.
- **Xarxa de difusió:** transporta el senyal audiovisual des del centre emissor fins a l'usuari o consumidor.

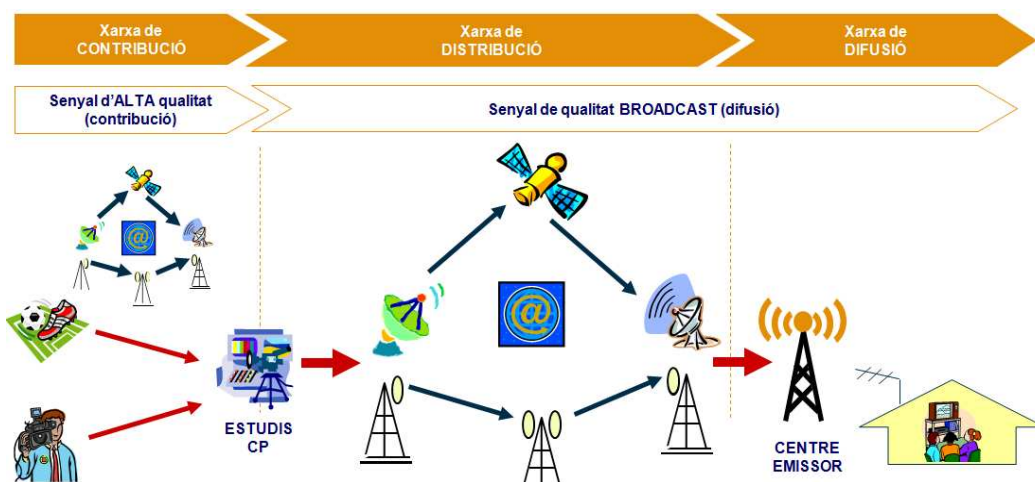


Fig. 1.1 Cicle del senyal de radiodifusió [2]

A la Fig 1.1 es mostra de manera esquemàtica el procés seguit pel senyal audiovisual des de la seva generació o obtenció fins al lliurament a l'usuari.

En l'actualitat, els senyals de televisió digital s'organitzen en canals múltiples de 8 MHz d'ample de banda numerats des del 21 (470 – 470 MHz) fins el 69 (854 – 862 MHz). A Espanya els canals disponibles són compartits pels serveis de difusió públics i privats de nivell estatal, autonòmic o nacional i local.

Una de les característiques de la televisió digital és que en un sol canal múltiple s'hi pot transportar el contingut de fins a quatre o cinc programes SD de manera simultània². Aquesta és la raó per la qual aquests canals són anomenats “canals múltiples” o “múltiplex”.

L'objectiu d'una capçalera digital és el de formar un senyal múltiple a partir dels senyals entregats en banda base pel proveïdor de continguts. El senyal generat s'entrega a la xarxa de difusió, que s'encarrega de fer-lo arribar a l'usuari final. Els agents que participen en el procés són:

- **CP** (Centre Proveïdor de Continguts): emplaçament on es generen els serveis audiovisuals i/o valor afegit (aplicacions interactives) que formen part del múltiplex.
- **CMUX** (Centre de Multiplexació): emplaçament on s'integren tots els serveis en un únic senyal: el múltiplex.
- **CE** (Centre Emissor): centre de difusió que transmet el senyal múltiplex a la seva regió de cobertura.

La capçalera digital consisteix doncs en tot l'equipament instal·lat al CP i al CMUX per a:

1. Codificar els continguts que entrega el CPP i generar serveis.
2. Multiplexar tots els serveis en un únic senyal múltiplex.
3. Adaptar el senyal múltiplex per a la seva distribució als CE que formen la xarxa de difusió.

Els processos que es duen a terme sobre el senyal es poden visualitzar esquemàticament en el dibuix de la Fig 1.2, on també s'identifica on té lloc cada procés (al CP o al CMUX).

² Els programes en HD ocupen l'amplada de banda equivalent a quatre en SD, de manera que només se'n pot transportar un en cada múltiplex.

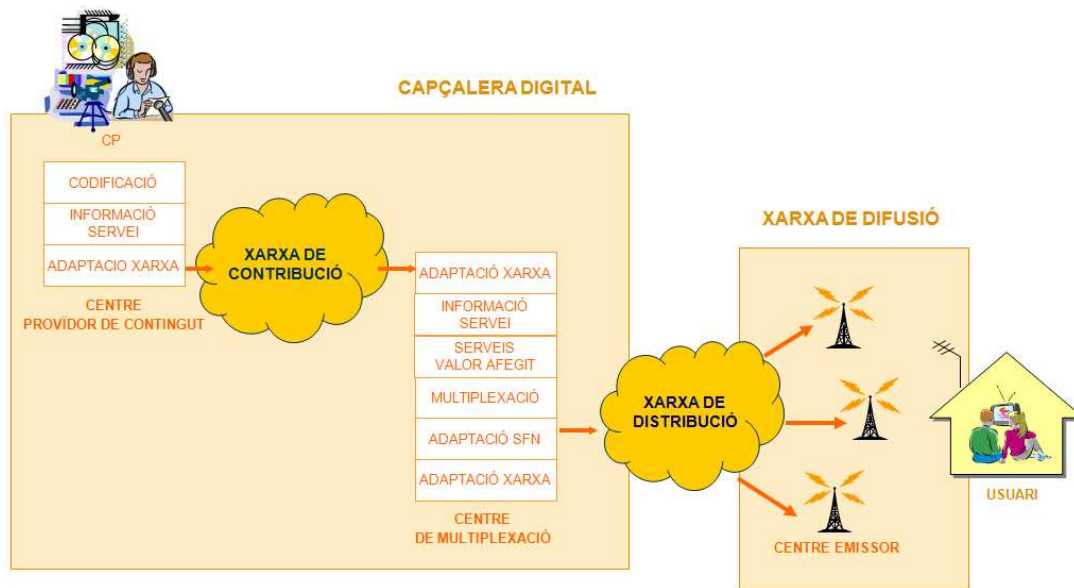


Fig. 1.2 Processos aplicats al senyal de radiodifusió [2]

1.1.2. DVB: *Digital Video Broadcasting*

És l'estàndard de la televisió digital europea (utilitzat també en altres parts del món), creat l'any 1993 per a donar resposta a la necessitat de definir un format comú que permetés la difusió de la Televisió Digital. Alguns dels aspectes que defineix són:

- Mètodes de modulació i correcció d'errors relacionats amb les transmissions terrestres (DVB-T/T2), per cable (DVB-C/C2) o per satèl·lit (DVB-S/S2).
- Algorisme comú per als sistemes d'enciptació i accés condicional (DVB-CA).
- Transmissió de la informació de servei (DVB-SI) que permet a l'espectador un accés senzill i ràpid als serveis o programes que es transporten.
- Utilització de l'estàndard de codificació MPEG-2 i H.264 per a la compressió dels senyals audiovisuals.

El consorci DVB en si no crea els estàndards, sinó que proporciona les especificacions que s'han de complir a organitzacions com la ETSI, CENELEC, ITU-R, ITU-T o DAVIC.

A la Fig 1.3 es mostra un diagrama de flux on es representen els diferents processos que tenen lloc sobre el senyal audiovisual digitalitzat i la seva informació associada.

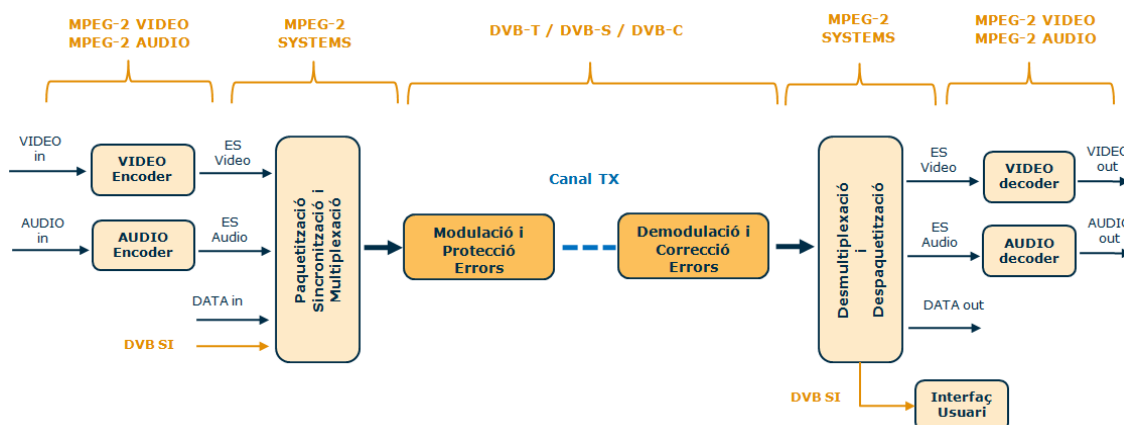


Fig. 1.3 Esquema de processos aplicats al senyal en transmissió i recepció [2]

1.1.3. Senyal audiovisual. MPEG-2 Systems

En cas que les càmeres de que es disposi siguin analògiques, el senyal es digitalitza als estudis. S'utilitza un sistema compatible amb les senyals analògics obtinguts de les càmeres.

- Norma ITU-R 601 / SMPTE 259M: senyal SD-SDI (270 Mbps): PAL (625/50) i NTSC (525/60)
- Norma SMPTE 292M: senyal HD-SDI (1.5 Gbps): 1080i i 720p
- Norma SMPTE 372M: senyal HD-SDI (3 Gbps): 1080p

Es requereix un gran ample de banda per al transport del senyal, fet que provoca que calgui comprimir-lo seguint els estàndards MPEG-2 o MPEG-4 Part 10/H.264 AVC.

Prèviament a la definició dels diferents formats de contenidor de MPEG-2 Systems és necessari conèixer el concepte de "programa". En televisió digital un programa és un senyal audiovisual constituït per un *stream* de vídeo, un o més *streams* d'àudio i altres *streams* corresponents a diferents fluxos de dades (teletext, dades privades, etc.).

Per a construir un programa cal associar i sincronitzar els diferents fluxos dels senyals que el formen (vídeo, àudio i dades). La secció "Systems" de MPEG-2, corresponent a l'estàndard ISO/IEC 13818-1, defineix dos formats de contenidor diferents però relacionats entre si:

- **MPEG TS** (*Transport Stream*): dissenyat per transportar senyals audiovisuals en canals hostils (amb alta probabilitat d'error). No es defineix amb precisió el principi ni el final de la informació audiovisual i,

per tant, està orientat a fluxos continus (com és el cas de la radiodifusió de televisió).

- **MPEG PS** (*Program Stream*): contenidor dissenyat per continguts amb un inici i un final determinats (pel·lícules, videoclips, etc.) que s'emmagatzemen o transmeten en canals menys hostils (discs durs, discs òptics, memòries *flash* o canals amb molta protecció contra errors).

La Fig 1.4 mostra els diferents fluxos de dades i la relació que tenen entre si.

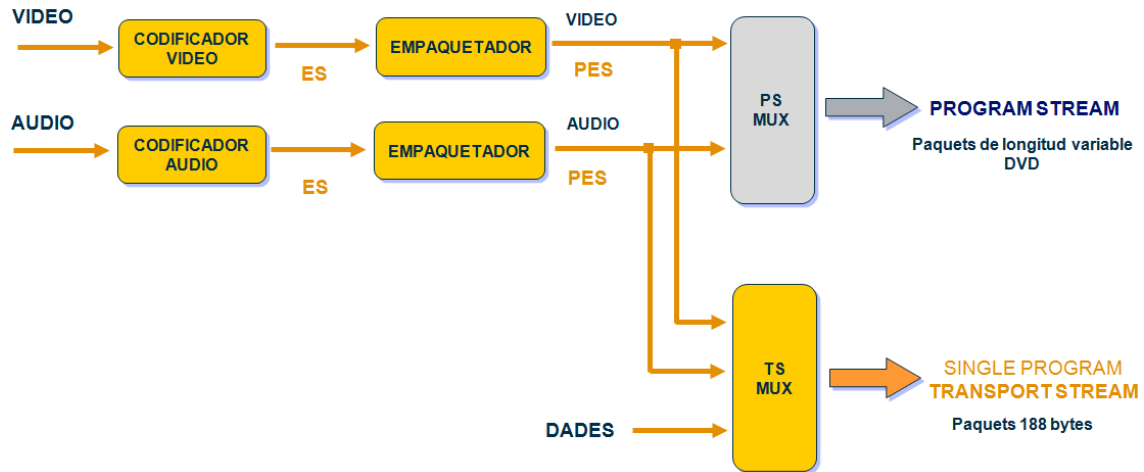


Fig. 1.4 Fluxos de dades de sistemes MPEG-2 PS i TS [2]

on els acrònims corresponen a:

- **ES** (*Elementary Stream*): flux continu de dades que conté informació d'una única font de vídeo o àudio. No conté informació de sincronització relativa a altres fluxos, però sí que pot transportar informació d'ordenació interna de les unitats d'accés (imatges de vídeo o trames d'àudio).
- **PES** (*Packet Elementary Stream*): empaquetat d'un ES. Els paquets estan formats per una capçalera i el *payload*, que és on es transporta la informació de l'ES. Afegeix marques temporals a les unitats d'accés ES per alinear-les temporalment i sincronitzar els diferents fluxos que formen part d'un programa. Cada PES s'identifica amb un PID (Packet Identifier) de 13 bits (rang 0-8191 en decimal) que permet posteriorment la seva desmultiplexació. Les unitats PES coincideixen amb les unitats d'accés ES (afegint la capçalera PES) i tenen longitud variable (normalment gran).
- **PS** (*Program Stream*): és el resultat de combinar en un sol flux un o més PES amb una base de temps comuna, a la qual estan referits els *timestamps* inserits a la capçalera PES. El flux PS va combinant els paquets PES de tots els fluxos audiovisuals que componen un programa, de manera que tots avancen en paral·lel i respecten els

requeriments temporals del descodificador per presentar la imatge i el so descodificats en els instants adequats.

- **TS (Transport Stream):** resultat de combinar un o més programes en un sol flux sota una base de temps comuna. El flux TS està compost per paquets de longitud fixa (188 bytes: 4 bytes de capçalera i 184 bytes de *payload*). La informació transportada a la capçalera és la següent:
 - **Bits 1..8: byte de sincronisme:** combinació fixa (0x47) que indica l'inici d'un paquet TS.
 - **Bit 9: Transport error indicator:** adverteix de la presència d'errors en el camí de retorn. No és rellevant en aplicacions de radiodifusió.
 - **Bit 10: Payload unit start indicator:** 1 indica que el paquet TS transporta l'inici de la capçalera PES d'una unitat d'accés (imatge o trama d'àudio).
 - **Bit 11: Transport priority:** 1: prioritari; 0: no prioritari. No és rellevant en aplicacions de radiodifusió.
 - **Bits 12..24: PID (Identificador de Paquet):** els 13 bits del PID, que identifica el flux ES del qual es transporten dades al *payload* del paquet TS.
 - **Bits 25..28: Adaptation field control:** indica la presència d'un camp de dades d'adaptació (que pot incloure timestamps o altres informacions).
 - **Bits 29..32: Continuity counter:** comptador de seqüència en el rang 0 – 15.

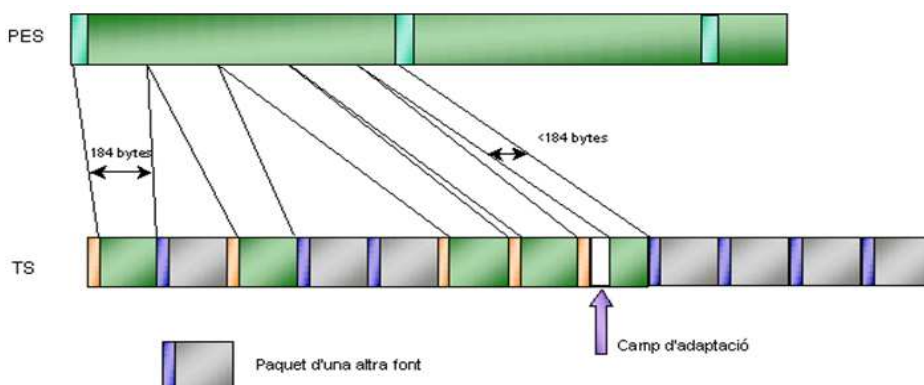


Fig. 1.5 Multiplexació de dades de varis PS en un TS [2]

La Fig 1.5 mostra gràficament la distribució de la informació de diferents PS en un sol TS.

Com que la taxa binària (*bitrate*) generada pels codificadors MPEG pot ser variable, però en canvi els paràmetres de modulació de DVB-T/S/C fan que el flux binari modulats sigui de taxa constant, és necessari introduir un *padding* o farciment per aconseguir el *bitrate* requerit pel multiplex. Alguns paquets TS sense informació útil, anomenats *null packets* i identificats amb el PID 8191 (1xfff), es multiplexen juntament amb els continguts audiovisuals i les taules PSI/SI.

1.1.4. Senyalització

1.1.4.1. Les taules PSI

Es defineixen quatre taules (taules PSI o "*Program Specific Information*") que faciliten al descodificador el procés de desmultiplexació:

- **PAT:** *Program Association Table*
 - PID = 0 (reservat per MPEG-2 *Systems*).
 - És la primera taula que es consulta al sintonitzar un nou TS. És per això que el valor del PID és fix.
 - Conté la llista dels programes que formen el TS i els PID dels seus corresponents PMT.
 - Sempre es defineix el programa #0 apuntant a la NIT amb PID = 16.
 - El seu contingut no pot encriptar-se.

- **PMT:** *Program Map Table*
 - Existeix una PMT per cada programa audiovisual.
 - Es transmet amb el PID indicat a la PAT (no són valors reservats per MPEG-2 *Systems*, sinó assignats arbitràriament).
 - Identifica i indica la localització dels *streams* que formen cada un dels serveis (programes) i del camp PCR (temporització).
 - El seu contingut no pot encriptar-se.

- **CAT:** *Conditional Access Table*
 - PID = 1 (reservat per MPEG-2 *Systems*).
 - Proporciona l'associació entre un o més sistemes d'encriptat (accés condicional) així com els seus fluxos de claus i permisos.
 - La informació és privada i depèn del sistema d'accés condicional. Inclou la localització dels EMM (*Entitlement Management Messages*).

- Els EMM contenen informació privada que indica els nivells d'autorització o els serveis pels descodificadors específics.
 - El contingut privat dels paquets que contenen els EMM i la resta d'informació d'accés condicional sol anar encriptat.
- **NIT: Network Information Table**
- PID = 16 (reservat per MPEG-2 Systems).
 - La seva informació és opcional, amb contingut privat i no definit per l'estàndard MPEG-2, però DVB en defineix els continguts i la fa obligatòria.

Les taules s'han de repetir periòdicament per a garantir el procés de descodificació. La PAT, per exemple, s'ha d'enviar cada 500 ms.

1.2. La xarxa *broadband*

Internet neix d'un projecte de recerca finançat pel ministeri de defensa nord-americà que té com a objectiu permetre la compartició d'informació científica i militar entre ordinadors. Es va posar en marxa l'any 1969 (amb el nom Arpanet) connectant quatre grans ordinadors localitzats en diferents universitats del sud-oest dels Estats Units. En els últims 40 anys ha evolucionat fins a convertir-se en una xarxa global d'ordinadors que connecta milions d'equips a través dels nodes de commutació i encaminament (*routers*).



Fig. 1.6 Esquema visual de la xarxa *broadband* [3]

Les principals característiques d'aquesta xarxa són les següents:

- Permet connectar xarxes d'ordinadors de diferents arquitectures.
- És una xarxa robusta: els paquets poden viatjar per camins diferents en funció de l'estat de l'enllaç en cada instant concret.
- És una xarxa descentralitzada i distribuïda.
- És escalable: funciona bé independentment del nombre d'equips que s'hi connectin, independentment de la xarxa local a la que pertanyin.

Degut a la seva notable complexitat, podem estructurar Internet en piles de protocols seguint el model facilitat per la pila OSI de la ISO (Fig 1.7).

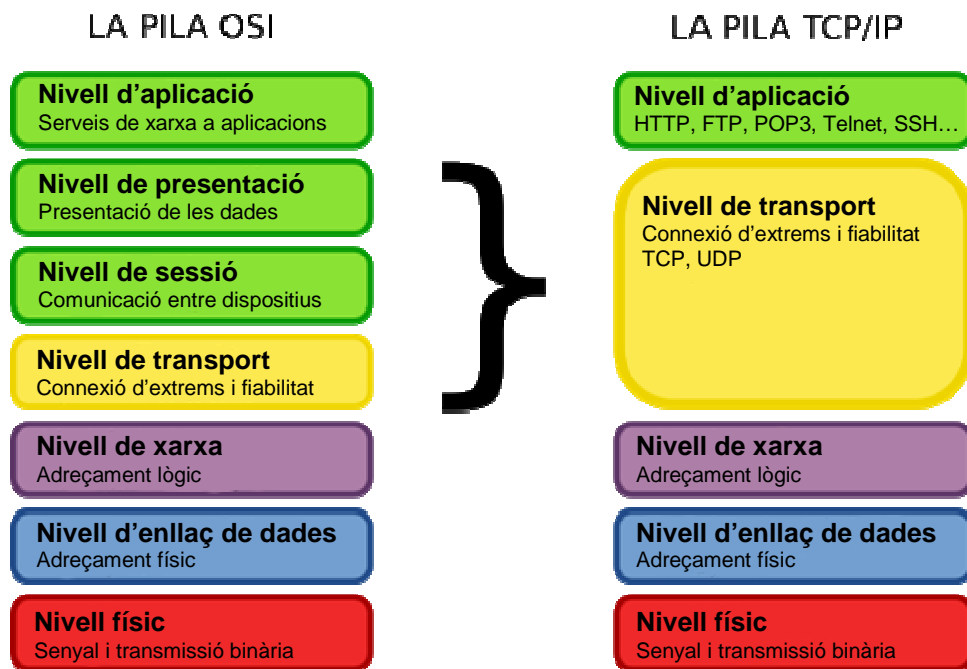


Fig. 1.7 Comparació entre la pila OSI de la ISO i la pila TCP/IP

1.2.1. Els protocols TCP/IP

Internet Protocol (IP, RFC 791) és el protocol de xarxa, encarregat del lliurament de paquets. No està orientat a connexió, cosa que significa que cada paquet es tracta de manera independent a la resta. No garanteix que el paquet arribi al seu destí ni tampoc que ho faci en un temps determinat.

Transmission Control Protocol (TCP, RFC 793) és el protocol de transport més comú a Internet juntament amb UDP (RFC 768). A diferència del protocol IP, TCP s'executa als extrems de la comunicació, està orientat a connexió i és fiable. Aquesta fiabilitat s'aconsegueix gràcies al mecanisme ARQ, que repeteix automàticament els fragments d'informació en funció d'un diàleg de confirmacions entre font i destinació. Aquest mecanisme de retransmissions pot introduir retards i provoca que no sigui apte per aplicacions en temps real, en les quals s'acostuma a utilitzar el protocol UDP, que no retransmet els fragments i per tant no és fiable.

1.2.2. El protocol HTTP

Al nivell d'aplicació de la pila TCP/IP hi trobem diversos protocols, entre ells HTTP (RFC 1945, 2616), FTP (RFC 959), POP3 (RFC 1939), Telnet (RFC 854), SSH (RFC 4252), etc.

HyperText Transfer Protocol (HTTP) és el protocol encarregat de la transferència de documents web. Utilitza els serveis TCP/IP com a mecanisme fiable de transferència de dades. Ofereix una forma de representar peticions del client i respostes del servidor web. Els tipus més comuns de petició són els següents:

- **HEAD**: petició per comprovar l'estat de la connectivitat client – servidor.
- **GET**: petició d'un document.
- **POST**: permet que el client faciliti informació al servidor. En el nostre cas concret, s'han utilitzat peticions POST per al formulari d'actualització de la base de dades.

També hi ha altres peticions: PUT, DELETE, TRACE, OPTIONS i CONNECT.

Els codis de resposta dels servidors es divideixen en:

- **1xx Informació**: petició rebuda, resposta provisional del servidor.
- **2xx Operació amb èxit**: petició rebuda i acceptada pel servidor.
- **3xx Redireccionament**: el client ha de redirigir la petició.
- **4xx Error de client**: petició amb errors.
- **5xx Error de servidor**: la petició sembla vàlida però el servidor no la pot atendre.

Per a identificar recursos a Internet, cada un d'ells té assignada una URL (*Uniform Resource Locator*) pròpia. La URL és un identificador format pel protocol de comunicació a emprar, la màquina on s'allotja el recurs i el directori específic dins aquesta màquina. En el nostre cas, la URL de l'aplicació HbbTV s'indica en un dels camps de configuració de la taula AIT.

1.3. L'estàndard HbbTV (*Hybrid broadcast broadband TV*)

En aquest apartat es fa una síntesi de la informació sobre l'estàndard HbbTV disponible al document ETSI TS 102 796 v1.1.1 [1]. Aquesta és la versió que implementen els dispositius HbbTV disponibles a data de juliol de 2012. Recentment s'ha publicat la nova versió 1.5 [4], que complementa l'anterior i afegeix noves especificacions, com ara:

- HTTP *adaptive streaming* (MPEG DASH) [5].
- Esquema d'enciptació comú per a permetre la utilització de múltiples tecnologies DRM.

- Accés a la informació de la taula EIT³.

A la Fig 1.8 es mostra l'esquema de la situació actual de l'estàndard HbbTV.

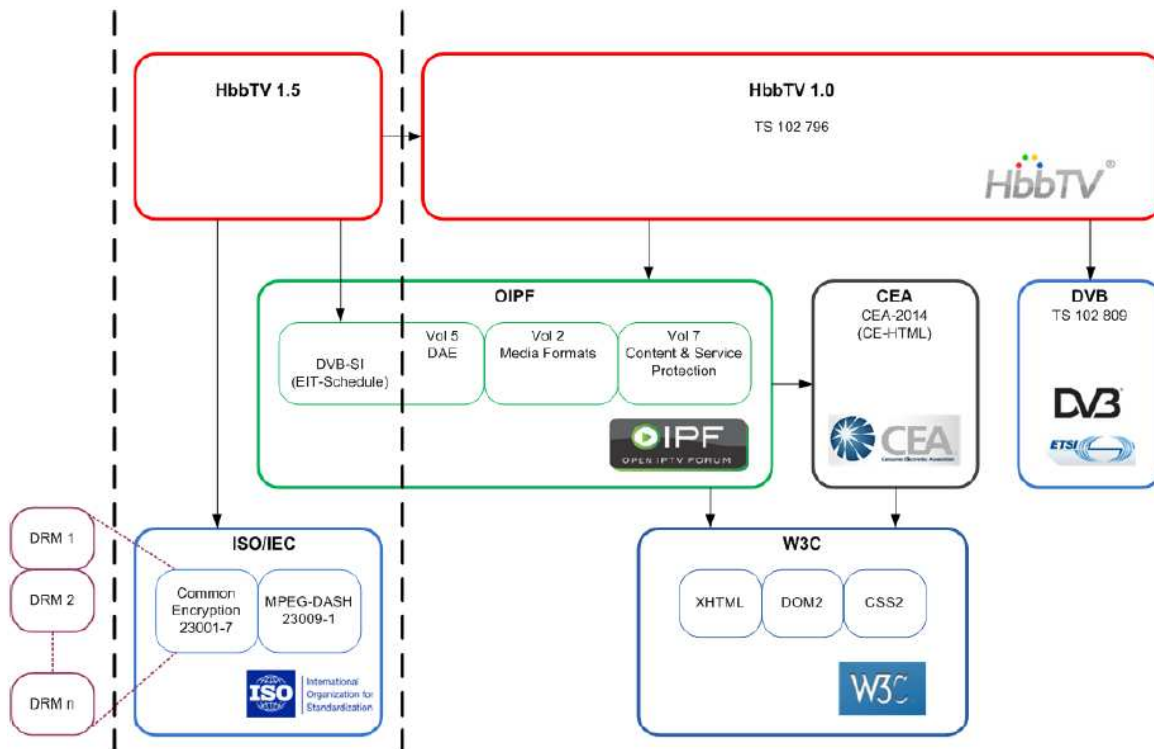


Fig. 1.8 Especificació HbbTV [4]

A continuació desglossarem la informació de la part corresponent a la primera versió de l'estàndard.

1.3.1. Introducció

Hybrid Broadcast Broadband TV és una iniciativa d'origen europeu que té l'objectiu de donar un valor afegit a les actuals tecnologies de la televisió digital, posant a l'abast dels radiodifusors una plataforma oberta per proporcionar serveis interactius i continguts sota demanda als consumidors finals. El factor clau es troba en la conjunció de dues tecnologies ja existents: la de la televisió digital (*broadcast*) i Internet (*broadband*).

El consorci industrial encarregat d'impulsar aquesta iniciativa inclou radiodifusors terrestres (com ara RTVE, TV3, Canal+...), instituts de recerca, operadors de satèl·lits (SES ASTRA, Eutelsat...), fabricants de terminals (LG, Philips, Panasonic, Sony...) i empreses encarregades de produir solucions software (ANT, Opera, ACCESS, Open TV...) [6]. A nivell català, podem

³ La taula EIT transmet informació sobre el conjunt d'events que es produeixen al múltiplex MPEG.

destacar la presència com a *supporters* d'empreses com Televisió de Catalunya i activa_multimèdia digital S.L., filial de la Corporació Catalana de Mitjans Audiovisuals. Com a membre del consorci HbbTV també destaca la presència de la multinacional catalana Abertis Telecom.

El gran avantatge de disposar d'un estàndard per aquesta tecnologia híbrida és el fet que arribarà a tothom qui disposi d'un terminal que la suporti, independentment de la marca o el model del receptor. D'aquesta manera s'aconsegueix un mercat potencial molt ampli. Al tractar-se d'una plataforma oberta, queda clar que HbbTV es troba en una situació privilegiada respecte competidors com poden ser els mercats tancats de televisió connectada (IPTV, Samsung SmartTV, etc.). És necessari remarcar que el fet que una televisió en concret tingui accés a un mercat tancat d'aplicacions proporcionat pel fabricant, no implica que no pugui accedir a les aplicacions HbbTV que li arriben senyalitzades via *broadcast*⁴.

1.3.2. Funcionament

Com s'ha comentat anteriorment, els terminals híbrids que suporten HbbTV tenen la capacitat de connectar-se a dues xarxes de manera simultània:

- La xarxa ***broadcast*** (DVB-T/T2, DVB-S/S2, DVB-C/C2): a través d'aquesta xarxa de televisió digital el terminal pot rebre els continguts de radiodifusió tal i com els entenem avui en dia. Aquests continguts reben el nom de continguts audiovisuals lineals (*linear A/V content*, en anglès). A més, a través d'aquesta xarxa també es rep informació de la senyalització de l'aplicació i, en alguns casos, dades de l'aplicació.
- La xarxa ***broadband*** (IP): el terminal es pot connectar a Internet a través d'una interfície *broadband* (típicament ADSL). Això permet establir una comunicació bidireccional amb els proveïdors d'aplicacions. És gràcies a aquesta característica que l'estàndard HbbTV fa possible la interactivitat usuari-aplicació. A través d'aquesta nova interfície es reben dades de l'aplicació i contingut audiovisual no lineal (*non-linear A/V content*, en anglès). En cas que el terminal no estigui connectat a Internet, el seu funcionament és exactament igual al dels dispositius receptors de TDT que coneixem.

La Fig 1.9 és un exemple esquemàtic on es mostra un terminal híbrid connectat a la xarxa de radiodifusió DVB-S i a Internet.

⁴ Un exemple és la Samsung SmartTV que hem utilitzat a les proves, que és capaç d'executar aplicacions propietàries Samsung, però també aplicacions HbbTV.

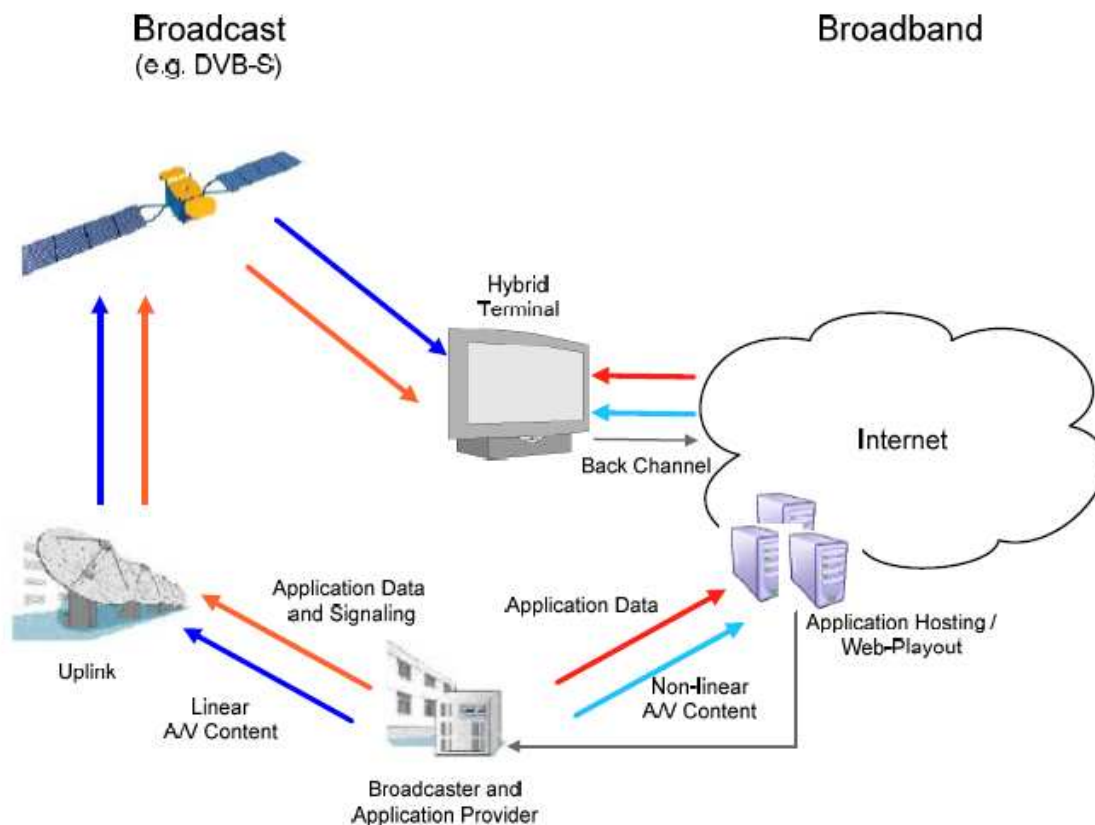


Fig. 1.9 Esquema d'una xarxa híbrida HbbTV [1]

1.3.3. Tipus d'aplicacions

Els serveis accessibles gràcies a HbbTV seran facilitats pels radiodifusors, els propis fabricants dels dispositius o altres proveïdors independents d'aplicacions. Es pot intuir doncs, que segons qui proporcioni el servei, aquest tindrà unes característiques diferents. La diferenciació principal és des d'on es llança l'aplicació (des d'un portal, des d'un canal en concret, des de diversos canals, etc.).

Segons defineix l'estàndard, les aplicacions es divideixen en dues categories:

- Aplicacions ***broadcast-related***: es declaren a la senyalització DVB i, tal com s'indica al seu nom, són aplicacions relacionades amb un servei de radiodifusió. Per tant, aquest tipus d'aplicació es llança des d'un o més canals de televisió. El contingut pot arribar via *broadcast* (parcialment inserit al *stream*) o via *broadband* (més comunament). Alguns exemples d'aplicacions *broadcast related* són:
 - Aplicacions de botó vermell
 - Serveis relacionats a un programa
 - Teletext digital (pot ser llançat amb el botó "txt" del comandament)

Les aplicacions de botó vermell o els serveis relacionats a un programa s'acostumen a senyalitzar com a *AUTOSTART*, de manera que la seva execució comença tan bon punt es sintonitza el canal al qual estan associades. Tot i així, s'acostuma a deixar a l'elecció de l'usuari el fet d'accedir o no als continguts no lineals amb un missatge previ d'acceptació.

- Aplicacions ***broadcast-independent***: es tracta d'aplicacions que només estan disponibles via *broadband* i que no es declaren en cap servei del TS. Els proveïdors d'aquestes aplicacions poden ser fabricants o altres empreses independents. Alguns exemples d'aquest tipus d'aplicació són:
 - Jocs *online* a través de la TV
 - Xarxes socials, compartició de fotografies, etc.

En aquest cas, l'escenari més comú a partir del qual es llancen les aplicacions és un portal disponible mitjançant algun botó del comandament a distància.

Aquesta separació en l'organització de les aplicacions, juntament amb el fet que les aplicacions *broadcast-related* han de ser declarades a la senyalització DVB, assegura als radiodifusors que al seu canal només s'hi presentaran les seves aplicacions.

1.3.4. Escenaris

Tal com s'ha explicat, les aplicacions es poden declarar a la senyalització, poden no estar declarades o bé es poden referenciar en un portal independent. Al mateix temps, la navegació entre diferents aplicacions és possible; és a dir, és possible executar una aplicació des d'una altra. A mode d'exemple, algunes vegades, una aplicació *broadcast-related* pot contenir un enllaç cap a una aplicació *broadcast-independent* per a ampliar el ventall de continguts i/o serveis disponibles.

El consumidor sintonitza diferents canals de televisió, cadascun dels quals pot tenir una aplicació *broadcast-related* associada (o no). En cas afirmatiu, l'aplicació es pot llançar automàticament (mode *AUTOSTART*) al sintonitzar el canal (per exemple, mostrant un ganxo o *hook*). És l'usuari doncs qui decideix si accedir o no als continguts interactius disponibles. Per entrar a l'aplicació tan sols cal prémer el botó vermell del comandament a distància un cop apareix en ganxo a la pantalla.

Una aplicació *broadcast-related* pot redirigir l'usuari cap a una altra aplicació *broadcast-related* o cap a una *broadcast-independent*. Aquest darrer tipus d'aplicacions no poden ser llançades directament mitjançant senyalització associada a un canal. Només s'hi pot accedir si es prové d'una altra aplicació o bé a través d'un portal independent, típicament dissenyat pel fabricant. Les característiques del portal, així com la navegació un cop s'hi ha accedit, queden fora de l'estàndard HbbTV ja que depenen únicament i exclusiva del fabricant.

1.3.5. Tecnologies utilitzades a HbbTV

L'especificació HbbTV està basada en estàndards ja existents. Així doncs, no es tracta d'un nou desenvolupament tècnic, sinó més aviat d'una conjunció específica de tecnologies ja disponibles. A la Fig 1.10 es mostra el resum de les tecnologies a partir de les quals es desenvolupa l'especificació, i que descriurem a continuació.

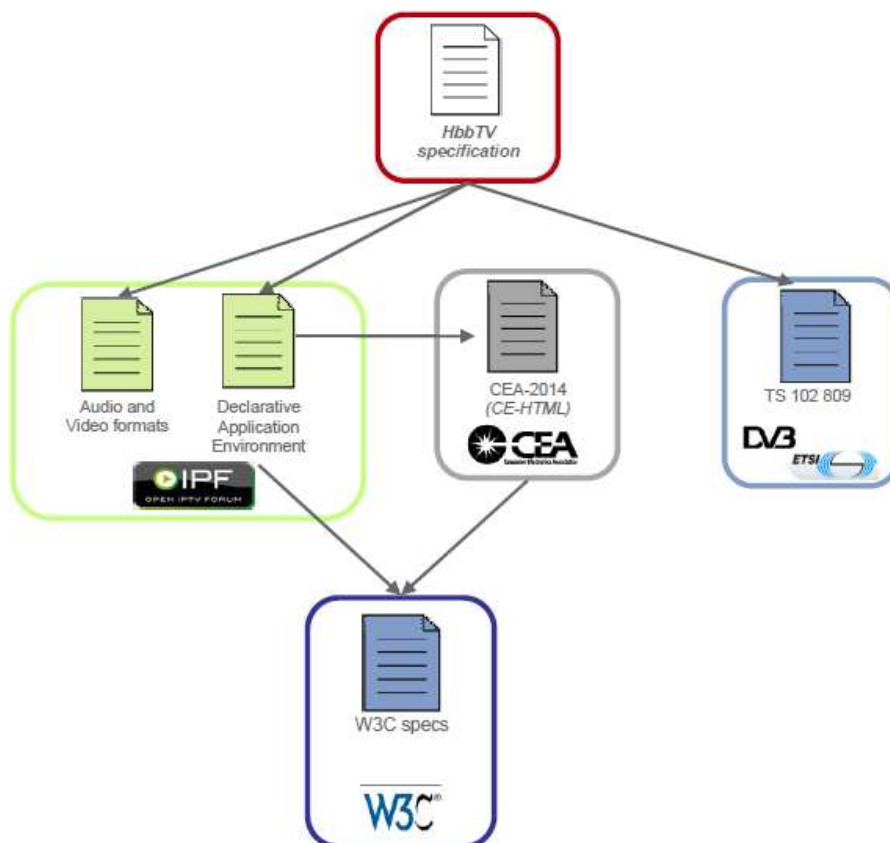


Fig. 1.10 Estàndards o tecnologies utilitzades en l'especificació HbbTV [1]

- **CEA – 2014** - *Web-based Protocol and Framework for Remote User Interface on UPnP Networks and the Internet (Web4CE)*, també conegut com a **CE-HTML** [7]: defineix les funcionalitats essencials del navegador. Està basat en els estàndards web del W3C i especifica un perfil HTML pels dispositius. Utilitza XHTML 1.0, DOM 2, CSS i JavaScript. També conté altres elements importants com per exemple la definició dels codis pels comandaments a distància de les televisions.

Alguns dispositius suporten XMLHttpRequest (conegut com AJAX), que permet actualitzacions dinàmiques de les dades de les aplicacions sense la necessitat de recarregar la totalitat de les dades.

- **Open IPTV Forum Release 1** [8-12]: aquesta especificació s'ha desenvolupat específicament per sistemes IPTV basats en l'estàndard DVB, però les APIs (JavaScript) que proporciona poden ser utilitzades en qualsevol sistema híbrid DVB, com per exemple HbbTV. Aquestes APIs, entre d'altres funcions, permeten combinar la imatge de la emissió en directe (*broadcast*) de la TV amb les pàgines HTML que defineixen l'aplicació.
- **ETSI TS 102 809** [13] "*Signalling and carriage of interactive applications and services in Hybrid Broadcast Broadband environments*": defineix la senyalització de les aplicacions HbbTV sobre *Transport Stream* per la seva difusió a través dels estàndards DVB. Aquestes aplicacions es poden executar en el context d'un servei específic en un múltiplex DVB. Això es fa a través de l'*Application Information Table* (AIT) en el servei DVB corresponent i s'indica al *Program Map Table* (PMT).

Per a la implementació de l'aplicació s'han utilitzat els llenguatges especificats al document CEA-2014 que són els indicats per l'estàndard HbbTV de la ETSI en el seu punt 4.4.

HTML

Sigles de *HyperText Markup Language*. Tal com indica el seu nom, es tracta d'un llenguatge de marcat, no de programació. És el llenguatge predominant per a l'elaboració de pàgines web.

La seva funció principal és la de descriure el contingut i l'estructura del text de l'aplicació, així com l'addició d'imatges. Es defineix una estructura de continguts molt essencial; no és útil per a dissenyar l'aparença gràfica. Tot i això, permet aplicar alguns estils gràfics senzills.

El llenguatge HTML s'escriu utilitzant etiquetes o *tags*, que permeten diferenciar els diferents elements i fer-ne referència de manera individual o col·lectiva al codi on es defineixen els estils gràfics de l'aplicació. L'estructura d'un element es compon de l'etiqueta inicial, el contingut i l'etiqueta final o de tancament. L'etiqueta inicial pot contenir atributs per a caracteritzar l'element. Als atributs és on es situen els identificadors per a dissenyar posteriorment el grafisme.

HTML és un llenguatge estàtic, és a dir, defineix els continguts de manera fixa i el navegador és capaç d'interpretar el codi i mostrar els resultats. Es pot trobar més informació sobre HTML a [14].

PHP

Llenguatge de programació utilitzat per a generar pàgines web i/o aplicacions dinàmiques. No s'executa al client, sinó que el llenguatge PHP és interpretat pel servidor, que a la seva vegada retorna la resposta en format HTML, de manera que la seva presència és transparent al navegador client. PHP doncs, ofereix el dinamisme no present a HTML, ja que permet obtenir els continguts canviants a partir de crides a una base de dades, per exemple.

El codi PHP es pot incloure dins el codi HTML entre les etiquetes `<?php` i `?>`. Tot el contingut tancat dins aquestes etiquetes serà traduït pel servidor prèviament al seu enviament al client conjuntament amb la resta de codi HTML. Es pot trobar més informació sobre PHP a [15-17].

CSS

Sigles de *Cascade Style Sheet*. És el llenguatge per a descriure els estils gràfics o la presentació dels elements definits al codi HTML.

Tot i que els estils es poden definir a l'etiqueta d'obertura de cada element mitjançant l'atribut *style*, és més senzill i més pràctic a l'hora de fer el manteniment de l'aplicació definir-los en un document CSS separat. A més, d'aquesta manera, es pot definir un estil comú per a un conjunt d'elements identificats amb el mateix atribut *class*.

La inserció d'un document CSS a un HTML es fa mitjançant la inclusió de la següent sentència a la capçalera:

```
<link rel = "stylesheet" type = "text/css" href = "doc.css">
</link>
```

Es pot trobar més informació sobre CSS a [18].

JavaScript (incloent AJAX)

Llenguatge de programació utilitzat per a crear pàgines web o, en aquest cas, aplicacions HbbTV dinàmiques que s'executa al client. Com a aplicació dinàmica entenem l'execució de diferents accions al generar diferents tipus d'events, permetent la interactivitat usuari - aplicació.

Una altra de les funcionalitats importants de JavaScript, i que com veurem més endavant serà d'utilitat per al desenvolupament de l'aplicació, és la possibilitat que ofereix de crear, llegir i eliminar *cookies*. Les *cookies* serveixen per emmagatzemar informació al navegador client amb l'objectiu de recuperar-la més endavant i poder discernir entre varies accions en funció de les dades obtingudes.

La inserció de codi JavaScript a un document HTML es pot fer de dues maneres: d'una banda, dins el cos del codi HTML podem inserir el codi JavaScript entre les etiquetes `<script></script>`. De l'altra, podem afegir un document JavaScript extern incloent la següent sentència a la capçalera HTML:

```
<script type="text/javascript" src="doc.js"></script>
```

Les sigles AJAX [19] provenen de *Asynchronous JavaScript and XML*. Aquesta tecnologia permet utilitzar l'objecte `XMLHttpRequest` per a intercanviar dades de manera asíncrona amb el servidor web mitjançant peticions HTTP i HTTPS. L'avantatge principal que ofereix és la

capacitat d'actualitzar determinades parts de l'aplicació sense necessitat de recarregar-la per complet.

Es pot trobar més informació sobre JavaScript a [20].

1.3.6. Aplicacions *broadcast-related*

Les aplicacions relacionades a un canal *broadcast* es poden trobar en un dels següents tres estats quan s'inicia la seva execució en mode *AUTOSTART*:

1. Mostrant el ganxo o *hook* d'accés.
2. No mostrant cap interfície d'usuari (només l'emissió *broadcast*).
3. Mostrant la interfície d'usuari completa.

En general, en les aplicacions que van associades a serveis de TV, l'estat inicial més comú és el primer, de manera que l'usuari és informat de la disponibilitat de continguts extra mitjançant un breu missatge. La resta de parts de l'aplicació no es mostren fins que es prem el botó vermell. Pel què fa a les aplicacions associades a un servei de ràdio, el més lògic és que l'estat inicial sigui el tercer, ja que l'execució i l'aparició automàtica de tots els elements de l'aplicació no suposa cap inconvenient en un procés de comunicació exclusivament auditiu.

Quan es produeix un salt des dels estats 1 o 3 cap al 2, l'aplicació s'ha d'encarregar de:

- Eliminar tots els elements gràfics de la pantalla.
- Parar la reproducció de vídeo o àudio provinent del canal *broadband*.
- Reprendre l'emissió del servei *broadcast* (si és que aquest s'ha aturat prèviament).
- Reescalar el vídeo a pantalla completa (si és que s'ha escalat prèviament).
- Activar l'àudio del *broadcast* (si s'ha desactivat amb anterioritat).
- Deixar d'escoltar qualsevol tipus d'event que no sigui prémer el botó vermell per tornar una altra vegada a l'estat 3.

Quan una aplicació canvia des de l'estat 2 al 1 o al 3, s'han de produir les següents accions:

- Mostrar els gràfics i els continguts de l'aplicació actualitzats.
- Informar al terminal quins són els events a escoltar en aquest nou estat

D'acord amb les definicions tècniques del cicle de vida d'una aplicació que es poden consultar al punt 6 del document ETSI TS 102 796 v1.1.1 [1], les aplicacions s'han de parar quan se'n llança una de nova o quan es produeix un canvi de canal. A més, es poden aturar per si soles com a resultat d'una acció de l'usuari (prémer el botó vermell) o degut a la seva lògica interna.

1.3.7. La taula AIT

Segons s'especifica en el punt 5.3.2 del document ETSI TS 102 809 [2], l'*stream* principal de la taula PMT d'un servei DVB que tingui una o més aplicacions HbbTV associades ha de fer referència a altres *streams* que permetin:

- Localització del *stream* que transporta la AIT (*Application Information Table*).
- Localització del/s *stream(s)* que transporten dades de l'aplicació (si es dóna el cas).

La taula AIT proporciona la informació necessària per a que el receptor pugui localitzar i executar les aplicacions relacionades a un servei de radiodifusió (veure Annex E).

1.3.8. Format dels continguts

L'estàndard no defineix els requisits dels formats de vídeo i àudio del canal *broadcast*. Aquests requisits estan definits en les especificacions corresponents a cadascun dels mercats on es desenvolupen els dispositius. Pel que fa als formats de la resta de continguts, els trobem definits a l'especificació OIPF *Media Formats*.

Taula 1.1. Formats de vídeo i àudio *broadband* [5]

System Format	Video Format	Audio Format	MIME Type
TS	AVC_SD_25 AVC_HD_25	HEAAC E-AC3	video/mpeg
MP4	AVC_SD_25 AVC_HD_25	HEAAC E-AC3	video/mp4

Com es pot veure són *codecs* de la família MPEG-4, incloent H.264/AVC per vídeo de definició estàndard (720x576 pixels) i d'alta definició (1920x1080) a 25 imatges/s, i codificadors d'àudio 5.1 del tipus *Dolby AC3* i *High-Efficiency Advanced Audio Coding*.

Taula 1.2. Formats dels continguts d'àudio pur [5]

Audio Format	MIME Type
MPEG1_L3	audio/mpeg
HEAAC	audio/mp4

En aquest cas també cal destacar l'acceptació del format *mp3*.

Taula 1.3. Formats de les imatges estàtiques [5]

Image Format	MIME Type
JPEG	image/jpeg
GIF	image/gif
PNG	image/png

1.3.9. Protocols de xarxa

Tots els protocols suportats per HbbTV són els que es defineixen a l'especificació OIPF Protocols v1.1 [6]. A continuació en farem un breu resum:

1.3.8.1 Protocols per streaming

L'estàndard suporta el protocol HTTP 1.1 per *streaming unicast*. El terminal ha de poder emmagatzemar dades per avançat equivalents a un màxim de 10 segons de reproducció, amb l'excepció que la velocitat de descàrrega sigui consistentment més baixa que la de consum. Per als continguts de vídeo MPEG4/AVC i àudio MPEG/AAC és opcional que els terminals suportin *streaming unicast* utilitzant RTSP i RTP.

1.3.8.2 Protocols de descàrrega

S'utilitza el protocol HTTP tant en mode de descàrrega persistent com de descàrrega progressiva, en la qual el contingut es pot començar a visualitzar prèviament a la finalització de la descàrrega.

1.3.8.3 Protocols de transport

El protocol HTTP tal com es defineix a RFC 2616 i l'HTTP sobre TLS definit als RFC 2818 i RFC 5246 (transport segur, encriptat i autènticat) han de ser suportats en el transport de les aplicacions per la xarxa *broadband*.

1.3.8.4 Capçalera HTTP User-Agent

Totes les peticions HTTP fetes per una aplicació HbbTV han d'incloure una capçalera `User-Agent` amb una estructura tal i com es descriu a continuació:

```
HbbTV/1.1.1 (<capabilities>; [<vendorName>]; [<modelName>];
[<softwareVersion>]; [<hardwareVersion>]; <reserved>)
```

La definició de cada un dels camps que formen la capçalera pot trobar-se al punt 7.3.2.4 del document ETSI TS 102 796 v1.1.1 [1].

CAPÍTOL 2. DISSENY D'APLICACIONS HbbTV

En aquest capítol s'introduiran les especificacions més essencials de les aplicacions DAE. Els conceptes es presenten de manera explicativa. Per a més detalls, consultar el document OIPF *Declarative Application Environment* [7].

Les especificacions del OIPF *Declarative Application Environment* (DAE) s'han de suportar tal i com es defineixen a l'annex A del document ETSI TS 102 796 v1.1.1 [1]. A més, es defineixen un conjunt d'APIs JavaScript específiques per HbbTV:

1. Gestionar *Stream Events* DSM-CC⁵
2. Accés als objectes DSM-CC mitjançant `XMLHttpRequest` (AJAX)
3. Extensions a l'objecte `application/oipfDrmAgent`

Entre aquestes tres APIs específiques, en l'aplicació desenvolupada ens hem centrat en l'adquisició i gestió de *Stream Events* per a notificar novetats a l'usuari i procedir a executar determinades accions davant de certs avisos.

2.1. Definició d'una aplicació DAE

Una aplicació DAE consisteix en un conjunt de documents (típicament JavaScript, CSS, i HTML) allotjats en un mateix FQDN⁶. L'aplicació DAE té accés a un objecte addicional anomenat `oipfApplicationManager` definit a la secció 7.2.1 del document OIPF DAE [7]. L'objecte `ApplicationManager` dóna accés a la classe `Application` definida a la secció 7.2.2 del mateix document, que proporciona propietats i mètodes JavaScript que són exclusius per a aquest tipus d'aplicacions, quedant al marge de les pàgines web tradicionals.

Les semblances entre una aplicació DAE i una pàgina web tradicional són evidents: en els dos casos hi ha un document inicial (normalment HTML) que inclou continguts d'altres documents com poden ser CSS, JavaScript, SVG, JPEG, PNG i GIF. El DOM, combinat amb `XMLHttpRequest` (AJAX), permet l'actualització de l'aplicació sense la necessitat de recarregar la totalitat de la pàgina.

La diferència principal entre una aplicació DAE i una pàgina web tradicional és el context en el qual aquesta es carrega i s'executa. El context inclou

⁵ *Digital Storage Media Command and Control*: canals de control associats als fluxos de dades de tipus MPEG-1 i MPEG-2. Utilitza el model client - servidor per sobre de la xarxa subjacent.

⁶ *Fully Qualified Domain Name*: inclou el nom de l'ordinador i el del domini associat a un equip en concret.

informació sobre l'estat d'una aplicació des del punt de vista de permisos, prioritats en cas d'insuficiència de recursos i altra informació similar.

2.2. Model d'aplicació DAE: cicle de vida

El cicle de vida d'una aplicació queda determinat per quatre factors:

1. El model d'aplicació.
2. El servei *broadcast* seleccionat i la seva pròpia selecció.
3. Les aplicacions senyalitzades al servei *broadcast* actual.
4. El codi de control de l'aplicació senyalitzada:
 - *AUTOSTART*
 - *PRESENT*
 - *KILL*
 - *PREFETCH*

Per a més informació sobre els codis de control, consultar la taula 3 del document OIPF DAE [7].

2.2.1. Creació d'una aplicació

L'aplicació HbbTV pot ser iniciada de les següents maneres:

- Directament per l'usuari a través dels botons del comandament a distància o pel menú proporcionat pel fabricant
- En resposta a la senyalització d'un servei *broadcast* en cas que aquesta tingui un codi de control del tipus *AUTOSTART*
- Per una aplicació que ja s'està executant (a través del mètode JavaScript `createApplication()`)

El document OIPF DAE, en el seu punt 5.1.1, mostra els detalls de les diferents maneres de generar una nova aplicació. Nosaltres ens centrarem en descriure la creació utilitzant l'API `Application.createApplication`.

L'objectiu de crear una aplicació s'aconsegueix mitjançant la creació d'un objecte `Application` amb el mètode `Application.createApplication`. Amb aquest mètode es crea una nova aplicació i s'afegeix en una determinada localització del model d'arbre definit a la secció 4.3.3 [7].

```
// L'objecte application/oipfApplicationManager té l'ID  
// "applicationmanager"
```

```
var appMgr = document.getElementById("applicationmanager");  
var self = appMgr.getOwnerApplication(Window.document);
```

Creem l'aplicació com a “filla” de l'aplicació actual:

```
var child = self.createApplication( url_of_application, true );
```

Cada aplicació té associat per defecte un objecte DOM del tipus `Window`. Inicialment es configura com a “*hidden*” per evitar efectes indesitjats durant l'inici de l'execució de l'aplicació. Un cop carregada la totalitat, es fa la crida al mètode `show()` de l'objecte `Application`. Podem conèixer quan s'ha carregat la totalitat de l'aplicació gràcies a l'event `onload`.

En el suposat cas que no sigui necessari que l'aplicació es vegi, l'objecte `Window` no es mostrarà mai. Això es pot aconseguir mitjançant CSS posant a l'element `BODY` la sentència `display: none`.

Tots els fitxers que formin part de l'aplicació s'han de trobar al mateix FQDN, és a dir, al mateix servidor. No és possible accedir a fitxers presents en un altre domini.

2.2.2. Finalització d'una aplicació

Per finalitzar una aplicació s'utilitza el mètode `destroyApplication()` tal i com es descriu a la secció 7.2.2.2 de l'OIPF DAE [7]. Hi ha la possibilitat d'afegir un *listener* per a que quan es produeixi un event del tipus `ApplicationDestroyRequest` es faci alguna acció prèvia a la destrucció. De la mateixa manera, quan s'accedeix a un enllaç extern (domini diferent) des de l'aplicació, aquesta s'ha de destruir de manera automàtica. Un cop executat el `destroyApplication()`, no es pot tornar a executar l'aplicació sense recarregar-la un altre cop.

Quan es destrueix una aplicació s'alliberen tots els recursos que aquesta estava utilitzant i totes les connexions actives. Això significa que qualsevol contingut audiovisual provinent d'Internet que s'està visualitzant en el moment de la destrucció també finalitza.

Per a més informació sobre els procediments seguits per les aplicacions *broadcast-related* davant d'un canvi de canal o d'una actualització de la senyalització, consultar el punt 5.2.3 del OIPF DAE [7].

A l'annex A es mostren dos diagrames de les accions realitzades al canviar de canal o davant un canvi en la senyalització de la taula AIT. També s'hi mostra un exemple del cicle de vida de les aplicacions, així com les transicions entre els seus diferents estats.

2.3. DSM-CC *Stream Events*

Els *Stream Events* són events *broadcast* transmesos sobre DSM-CC que permeten enviar una senyal a una aplicació interactiva en un moment determinat. Faciliten la conjunció entre el comportament de l'aplicació i l'emissió *broadcast*.

La informació detallada de l'API es pot trobar al punt 8.2.1 del document ETSI TS 102 796 v1.1.1 [1]. El terminal ha de suportar events del tipus “*do-it-now*” tal i com es defineix al punt 8 del ETSI TS 102 809 [2].

L'adquisició de *Stream Events* es gestiona amb els següents mètodes JavaScript:

```
void addStreamEventListener(String targetURL, String
eventName, EventListener listener)

void removeStreamEventListener(String targetURL, String
eventName, EventListener listener)
```

Un event del tipus `StreamEvent` es defineix de la següent manera:

```
interface StreamEvent : Event {
  readonly attribute String name;
  readonly attribute String data;
  readonly attribute String text;
  readonly attribute DOMString status;
}
```

Així doncs, un event en concret queda definit per les propietats “name”, “data”, “text” i “status”.

Per a gestionar els events generats utilitzem l'event intrínsec `onStreamEvent`, que en DOM 2 correspon a l'event `StreamEvent`.

CAPÍTOL 3. DISSENY I IMPLEMENTACIÓ D'UNA APLICACIÓ HbbTV PER A UN EVENT ESPORTIU

En aquest tercer capítol s'explica el desenvolupament de l'aplicació HbbTV que s'ha implementat. Primer s'introdueixen els requisits de funcionalitat generals i es fa una descripció de l'entorn de desenvolupament. Seguidament es detallen les característiques tant de l'aplicació HbbTV com del *webportal* d'actualització de continguts.

3.1. Requisits

L'objectiu d'aquest treball és implementar una aplicació HbbTV per a generar contingut audiovisual de valor afegit aplicada a un esdeveniment esportiu. Més concretament, el disseny i la lògica de l'aplicació està pensada per a events en els quals participin dos equips, com poden ser, per exemple, partits de futbol.

Els principals requisits de funcionalitat des del punt de vista de l'usuari són els següents:

- Accés a repeticions de jugades d'interès des de diversos punts de vista.



Fig. 3.1 Visualització de repeticions des de varis punts de vista

- Accés a informació útil de l'esdeveniment. En el cas d'un partit de futbol, poden ser estadístiques, alineacions dels dos equips, etc.
- Capacitat per escollir el millor jugador del partit i visualitzar els resultats de la votació.
- Capacitat per a participar activament mitjançant comentaris emesos a Twitter que continguin un determinat *hashtag*.

Així doncs, es pretén captar l'interès de l'usuari no només posant continguts addicionals al seu abast, sinó també aprofitant l'enllaç bidireccional que ofereix Internet per a que aquest pugui interactuar amb l'aplicació.

Dels requisits fonamentals a nivell d'usuari se'n deriven els requisits tècnics de l'aplicació. A mode de resum, aquests són els següents:

- Implementació i gestió d'una base de dades que reculli tota la informació que s'entregarà a l'usuari: llista de jugades amb la seva informació associada, estadístiques del partit, votacions dels millors jugadors, etc.
- Implementació d'un *webportal* per a actualitzar la base de dades amb noves jugades o informació recent del partit amb l'objectiu d'accelerar el procés per a assolir un important grau d'immediatesa de publicació.
- Implementació d'un *listener* de *Stream Events* per a poder avisar als usuaris de la disponibilitat de noves jugades a la llista mitjançant el canal *broadcast*.
- Disseny i implementació d'un codi d'aplicació que faci les crides corresponents a la base de dades per obtenir la informació, carregui la resta de continguts i finalment els organitzi de manera adequada amb l'objectiu de facilitar la interacció usuari-aplicació. A més, s'han de gestionar les funcionalitats específiques de les aplicacions HbbTV.

3.2. Arquitectura i entorn de desenvolupament

3.2.1. Arquitectura software

L'arquitectura de l'aplicació està basada en el patró de disseny conegut com a Model Vista Controlador (MVC), a partir del qual se separen les dades, la interfície d'usuari i la lògica de la pròpia aplicació en mòduls diferents [21].

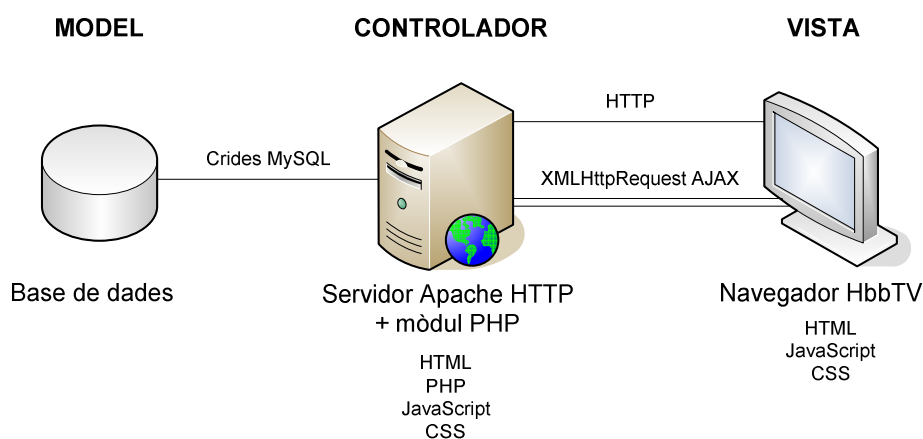


Fig. 3.2 Model Vista Controlador

El **model** fa referència a la informació amb la qual es treballa. En aquest cas, correspon a la base de dades i el seu sistema de gestió. Hem optat per utilitzar una base de dades de tipus relacional, degut principalment a la major presència que tenen al mercat actualment, molt per sobre que les no relacionals. Tot i la tendència a l'alça de les bases de dades no relacionals degut a les seves millores en escalabilitat, s'ha considerat que per a l'aplicació específica que es vol dissenyar aquesta no suposava un punt crític, almenys en una primera versió del desenvolupament.

S'ha escollit MySQL degut principalment a la facilitat d'interacció amb el codi PHP per a gestionar l'intercanvi de dades entre la pròpia base de dades i el servidor HTTP [22].

La **vista** presenta la informació del model de manera adequada, facilitant la interacció a partir d'una interfície d'usuari senzilla i amigable.

El **controlador** respon als events generats a partir de les diferents accions de l'usuari o de la lògica intrínseca de l'aplicació. Aquestes respostes inclouen peticions al model i a la vista.

La instal·lació dels diferents mòduls del model i el controlador es realitza en màquines físicament separades. En el nostre cas, per a simplificar les tasques de desenvolupament, s'ha allotjat tot en una mateixa màquina. Per la senzillesa que suposa, s'ha optat per instal·lar un pack anomenat *WampServer* [23] que inclou un servidor *Apache*, el mòdul PHP i MySQL. A més, disposa d'una eina per a la gestió de MySQL a través d'un navegador web anomenada *phpMyAdmin* [24]. Amb aquesta eina es pot realitzar qualsevol acció relacionada amb la base de dades: crear i eliminar bases de dades, crear i eliminar taules, modificar-les afegint o eliminant camps, executar sentències SQL, etc.

3.2.2. Entorns d'execució durant el desenvolupament

L'aplicació s'ha desenvolupat sobre emuladors de navegadors de dispositius HbbTV, de manera que no fos necessari un accés permanent a cap dispositiu físic. Posteriorment, un cop l'aplicació s'ha donat per acabada, l'hem provat amb l'equipament real.

A continuació descrivim els entorns d'emulació utilitzats.

3.2.2.1. *Fire HbbTV*

FireHbbTV [25] és un *plugin* que s'integra amb el navegador Firefox emulant el comportament d'un receptor i que permet el desenvolupament d'aplicacions HbbTV. Les APIs HbbTV s'activen automàticament al activar el *plugin*. La navegació per l'aplicació es pot fer utilitzant el teclat gràcies a l'emulació d'un comandament remot que ofereix aquesta eina (Fig 3.3).

Un dels principals punts forts de FireHbbTV és que l'objecte video/broadcast suporta el mètode `addStreamEventListener()` per a simular DSM-CC *Stream Events* (Fig 3.4), característica que no ofereix cap altra eina disponible en l'actualitat. Això ens ha permès simular l'emissió d'un *Stream Event* per actualitzar automàticament la llista de jugades disponibles a l'aplicació mitjançant AJAX. Podem controlar les respostes de l'aplicació en funció del camp "Data" de l'event.



Fig. 3.3 Emulador de comandament a distància de FireHbbTV



Fig. 3.4 Finestra de simulació de *stream events*

Es pot trobar més informació sobre aquest *plugin* a la referència [26].

A l'annex H es mostren captures de pantalla de l'aplicació executada des del navegador FireHbbTV.

Tot i que l'eina de simulació d'execució utilitzada per al desenvolupament de l'aplicació ha estat FireHbbTV degut principalment a la seva capacitat de gestionar *Stream Events*, també s'han utilitzat altres entorns per a comprovar la compatibilitat amb altres navegadors que es troben actualment al mercat de la televisió connectada.

3.2.2.2. *Opera TV Emulator*

Aquest emulador proporcionat per Opera no és específic per a aplicacions HbbTV, sinó que la seva funcionalitat engloba tot el conjunt d'aplicacions per a televisió híbrida desenvolupades amb HTML5 o CE-HTML.

Consisteix en una màquina virtual amb sistema operatiu Linux. Un cop instal·lada, proporciona un comandament remot virtual per a poder navegar per l'aplicació. A més, simula la limitació de memòria RAM d'un entorn d'execució real.

A l'annex H es mostren captures de pantalla de l'aplicació executada des de l'entorn Opera TV Emulator.

Seguidament es mostren els dos dispositius HbbTV que s'han utilitzat per a fer les proves reals.

3.2.2.3. *Engel EN2000*

També hem disposat d'un entorn d'execució realista amb el *set-top box* EN2000 del fabricant Engel, que permet executar aplicacions no senyalitzades a partir de la introducció de la seva URL mitjançant un document de text emmagatzemat en un *pen drive*. Cal destacar que per a que aquesta funcionalitat estigui disponible, el dispositiu ha d'estar rebent senyal de l'aire (hem de sintonitzar un servei de radiodifusió).



Fig. 3.5 Receptor EN2000 del fabricant Engel

3.2.2.4. Samsung SmartTV

Aquesta televisió connectada disposa d'un portal propi del fabricant on es poden instal·lar aplicacions exclusives de la marca. A més, també és compatible amb l'estàndard HbbTV, de manera que ha estat útil per a fer les proves de l'aplicació.



Fig. 3.6 Televisió Samsung SmartTV

3.3. Disseny i implementació

Aquest apartat es centra en l'explicació de l'aplicació desenvolupada i de la metodologia d'actualització de continguts a través del *webportal*.

Per a l'edició dels documents HTML, PHP, CSS i JavaScript s'ha utilitzat l'editor *open source* multi plataforma *Komodo Edit 7* [27] de l'empresa ActiveState.

3.3.1. Aplicació HbbTV

En aquest subapartat s'explica d'una banda el funcionament de l'aplicació des del punt de vista lògic (Fig 3.7) i, de l'altra els principals problemes que han sorgit durant el desenvolupament.

3.3.1.1. Lògica de l'aplicació

L'aplicació HbbTV implementada es divideix en dos documents principals que són els que en defineixen els dos grans blocs:

- **principal.php**: és el document PHP on es defineixen l'estructura i funcionalitats de la pàgina principal de l'aplicació. En aquesta pàgina es poden consultar les alineacions, les estadístiques i la llista de jugades disponibles (veure Fig 3.7, Fig 3.8).

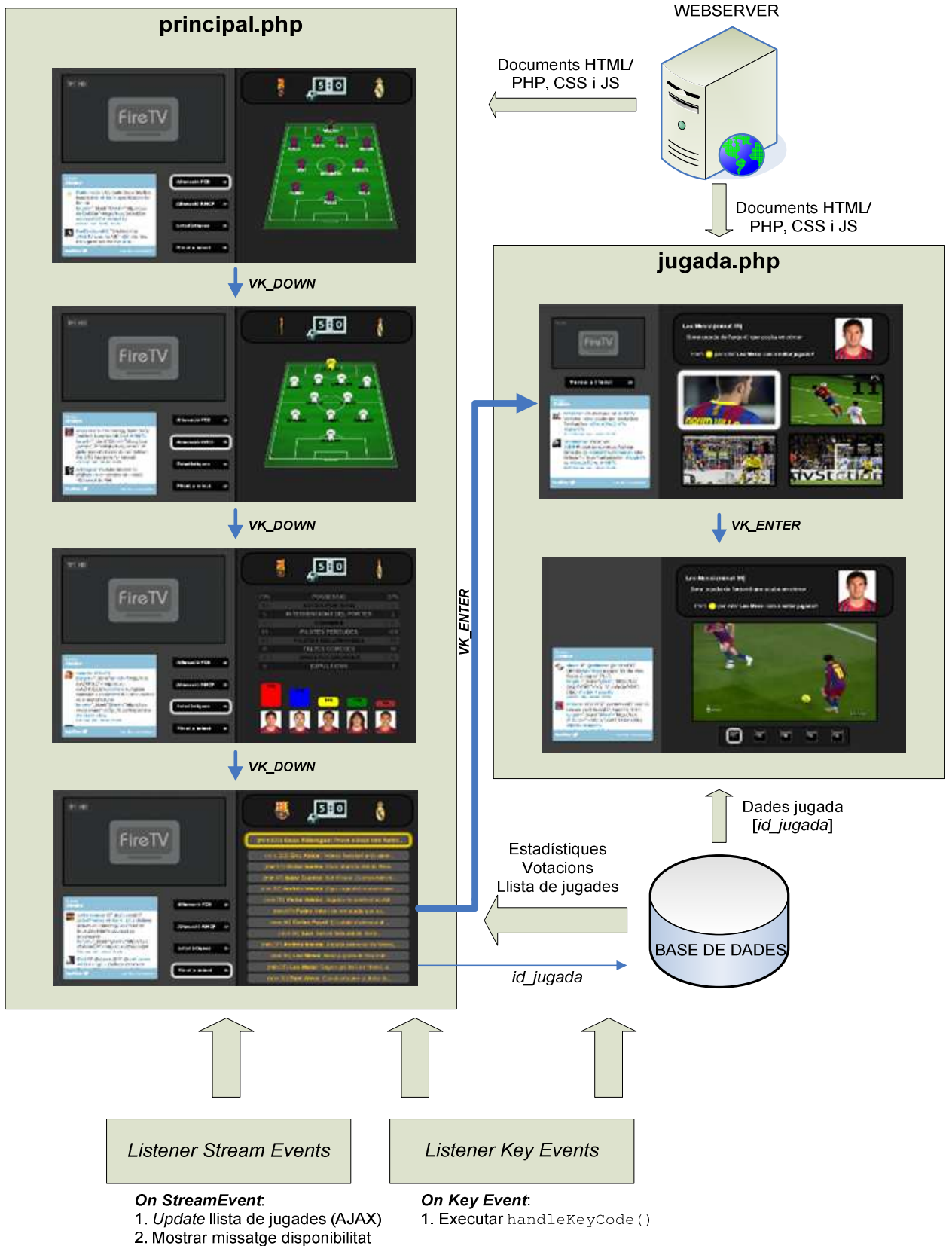


Fig. 3.7 Funcionament lògic de l'aplicació

En aquesta pàgina hi ha implementats dos *listeners*: el primer s'anomena `registerKeyEventListener()` i és el que s'encarrega de capturar els events produïts amb les tecles del comandament a distància. El segon és l' `addStreamEventListener(targeturl, eventname, listener)`, encarregat de registrar els *Stream Events* que es rebin pel canal *broadcast* DSM-CC.

A l'accedir a la pàgina principal es fan dues crides a la base de dades:

- `callEstadístiques()`
- `updateJugades()`

La primera crida fa una consulta a la taula "estadístiques" per obtenir els valors dels diferents camps a mostrar (veure Fig 3.8). A més, també es fa una consulta a la taula "votacions" per obtenir els resultats de la votació a millor jugador del partit que es fa entre els espectadors o usuaris de l'aplicació.

A l'executar la segona crida es crea un nou objecte `XMLHttpRequest()` amb el nom `xmlhttp`. Un cop creat, s'executen les següents sentències:

```
xmlhttp.open("GET", "getJugades.php", true);  
xmlhttp.send();
```

Amb el mètode `open` es fa un GET de les dades de les jugades obtingudes a l'executar el document "getJugades.php". Aquestes dades són les que es mostraran a la llista de jugades (minut, jugador protagonista i descripció).

La funció `updateJugades()` s'executa cada cop que es rep un *Stream Event* del tipus "nova jugada disponible".

El fet d'actualitzar la llista a partir de la creació d'un objecte `XMLHttpRequest()` evita que s'hagi de recarregar la totalitat de la pàgina principal en cada actualització, permetent estalviar ample de banda i refrescar dades de manera asíncrona.

Altres accions realitzades cada cop que accedim a la pàgina principal són:

- Accedir als documents JavaScript i CSS associats.
- Iniciar l'aplicació amb el mètode `initApp()`.
- Activar i escalar l'emissió *broadcast*.
- Situar el focus al primer botó (Alineació FCB) i realitzar les accions associades a aquest event `onfocus()`.

A tots els elements HTML que formen part de l'aplicació se'ls ha assignat el tipus de posicionament absolut, de manera que la seva localització es defineix a partir dels elements *left* i *top*. Per al disseny gràfic s'ha seguit el model de caixes tal i com s'explica a [34].



Fig. 3.8 Captura de la pantalla principal de l'aplicació amb FireHbbTV

- **jugada.php**: al prémer *ENTER* sobre una de les jugades de la llista s'executen les següents sentències:

```
//Obtenim l'id_jugada de la jugada amb el focus
var id_jug = document.getElementById('lisel').className;

document.location.href = 'jugada.php?id='+id_jug;
```

D'aquesta manera obtenim el número amb el que s'identifica la jugada a la base de dades i el passem a la pàgina *jugada.php* com a paràmetre "id" a la pròpia URL (relativa).

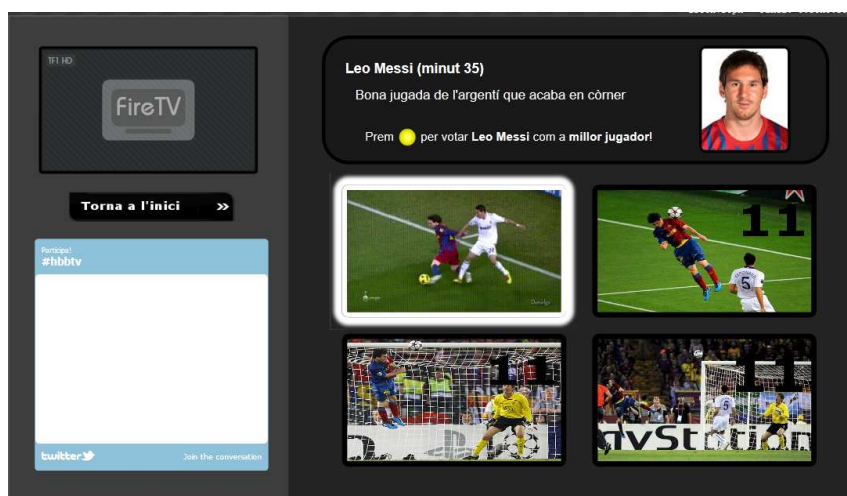


Fig. 3.9 Captura de la pantalla "jugada" de l'aplicació amb FireHbbTV

Així doncs accedim a la segona pàgina de l'aplicació, on es mostren les dades relacionades amb la jugada seleccionada i s'ofereix la possibilitat de visualitzar els vídeos de les repeticions (veure Fig 3.9). Podem recuperar l'identificador de jugada amb la següent sentència:

```
$id = $_GET['id'];
```

Amb aquest identificador fem una crida a la funció `callJugada($id)` per a establir una connexió amb la base de dades i obtenir tota la informació referent a la jugada seleccionada de la taula "jugades".

Altres accions que es realitzen a l'entrar a la pàgina d'una jugada són:

- Accedir als documents JavaScript i CSS associats.
- Activar el listener `registerKeyListener()`.
- Iniciar l'aplicació amb el mètode `initApp()`.
- Escalar l'emissió *broadcast*.
- Situar el focus al primer vídeo i realitzar les accions associades a aquest event `onfocus()`.

Com es pot veure a la Fig 3.9 en aquesta pàgina és on es permet votar al millor jugador del partit (prement el boto `VK_YELLOW`). Tal com s'ha definit la metodologia de votació, només es poden votar jugadors protagonistes d'alguna de les jugades de la llista de la pàgina principal.

A l'emetre un vot s'executa la funció `updateVotacions()`, des d'on s'executa el document "update_votacions.php".

3.3.1.2. Aspectes problemàtics

Durant el desenvolupament del codi de l'aplicació hi ha hagut alguns aspectes la solució dels quals ha requerit un cert temps de maduració de la idea en qüestió:

- **Visualitzar l'emissió broadcast i reproduir un vídeo a la vegada**
La idea principal era que a l'accedir a una jugada es mantingués l'emissió *broadcast* (tal com es fa ara) i, a més, poguéssim iniciar la reproducció d'un vídeo *mp4* servit per la xarxa *broadband* (per exemple, el corresponent a la imatge que té el focus).

Això no ha estat possible degut a que els receptors de què disposem només inclouen un descodificador de vídeo, i per tant o bé mostren la transmissió *broadcast*, o bé un vídeo rebut per xarxa IP. Ens hem hagut de conformar amb generar un GIF animat per a cada vídeo, de manera que al posar el focus sobre una jugada es mostri una previsualització semblant a una reproducció de vídeo.

- **Adquisició de *Stream Events***

Amb la descripció de la inclusió de *Stream Events* que es pot consultar a l'estàndard HbbTV [1] no en teníem prou per a saber com implementar les funcions necessàries per a la seva gestió.

A causa de les escasses referències bibliogràfiques respecte a aquest punt, s'ha hagut de recórrer a l'exemple proporcionat per l'aplicació de mostra *MIT-xperts HbbTV testsuite* [28]. Hem estudiat el codi corresponent a les proves amb *Stream Events* de l'apartat *StreamEvent* i l'hem adaptat a la nostra aplicació.

- **Sistema de votació del millor jugador**

El sistema de votació a millor jugador s'ha realitzat sense excessius problemes seguint l'exemple proporcionat per [29]. La qüestió més delicada ha estat el com limitar el número de cops que pot votar un mateix usuari. S'han contemplat diverses alternatives:

- Registrant la seva adreça IP.
- Registrant algun identificador del dispositiu físic, com ara l'adreça MAC (Ethernet) o similar.
- Amb un identificador de sessió HTTP.
- Utilitzant *cookies* [16].

Finalment s'ha optat per l'única solució viable: les *cookies*. Les altres opcions s'han descartat per diverses raons:

- a) L'adreça IP no es manté fixa per als dispositius que utilitzen DHCP.
- b) No s'ha trobat cap possibilitat d'accedir a un identificador únic del dispositiu amb llenguatge JavaScript.
- c) L'identificador de sessió es mantindria només durant la sessió actual; al tancar i tornar a executar l'aplicació es podria tornar a votar.

Per a gestionar les votacions mitjançant *cookies* s'han utilitzat les funcions que es detallen a l'annex F. Més endavant es veurà que no tots els dispositius HbbTV accepten aquest tipus de fitxer.

- **Inclusió de Twitter**

La idea principal era poder escriure i enviar comentaris amb un *hashtag* determinat mitjançant la pròpia televisió o dispositiu HbbTV. Això hagués implicat la programació d'un teclat virtual (mitjançant les tecles numèriques) i l'autenticació prèvia del compte d'usuari. Degut a que no són dos aspectes trivials i el temps disponible per al desenvolupament del treball és limitat, s'ha optat per eliminar aquesta funcionalitat i

incloure tan sols una finestra de Twitter on es vagin mostrant de manera successiva els missatges que contenen un determinat *hashtag*.

Això s'ha pogut fer gràcies a un *widget* proporcionat per Twitter. El codi a partir del qual s'edita es pot consultar a l'annex G.

3.3.2. Webportal d'actualització de continguts

3.3.2.1. Requisits

L'objectiu del *webportal* és posar a l'abast de la persona encarregada dels continguts de l'aplicació una eina útil per a gestionar i actualitzar la base de dades de manera senzilla i eficaç.

El portal serà un formulari on es recullen tots els camps d'estadístiques del partit per a poder-los actualitzar. A més, s'implementa també la funcionalitat de pujar una nova jugada. Aquest és un procediment més laboriós:

- Afegir les dades relacionades amb la nova jugada en un formulari.
- Pujar els quatre vídeos amb les diferents perspectives de la jugada al servidor.
- Generar les imatges JPG i els GIF animats corresponents a cada un dels vídeos.

El portal s'ha implementat amb HTML / PHP.

3.3.2.2. Disseny i configuració

A la Fig 3.10 es mostra una captura de pantalla del *webportal*:

HbbTV Formulari de gestió de continguts

Estadístiques de l'equip LOCAL	Estadístiques de l'equip VISITANT
GOLS 5	0 GOLS
POSSESIÓ 73	27 POSSESIÓ
XUTS 15	5 XUTS
ATURADES 5	2 ATURADES
CORNER 4	5 CORNER
PÈRDUES 95	103 PÈRDUES
RECUPERACIONS 85	76 RECUPERACIONS
FALTES 11	16 FALTES
GROGUES 2	5 GROGUES
EXPULSIONS 0	1 EXPULSIONS

Nova jugada

Minut del partit:

Protagonista: Foto:

Descripció de la jugada:

Adjuntar vídeos

Cap fitxer seleccionat.

Cap fitxer seleccionat.

Cap fitxer seleccionat.

Cap fitxer seleccionat.

Fig. 3.10 Captura de la pantalla principal del *webportal*

Veiem dos sectors diferenciats: mentre que a l'esquerra es mostren les estadístiques del partit, a la dreta hi ha el formulari amb els diferents camps necessaris per a afegir una nova jugada a la llista.

Quan l'usuari entra al portal es fa una crida automàtica a la base de dades per obtenir les estadístiques actuals del partit i mostrar-les en els corresponents camps *input-text*.

Al prémer el botó "Actualitzar" es fa una actualització dels diferents camps d'estadístiques de la base de dades.

Si volem inserir una nova jugada, cal que emplenem els camps corresponents i que adjuntem els quatre fitxers de vídeo *mp4* corresponents [30, 31]. El fet d'adjuntar fitxers de vídeo provoca que la sintaxi de la capçalera de l'element `<form>` corresponent a "Nova jugada" sigui la següent:

```
<form action = "nova_jugada.php" method = "post" enctype = "multipart/form-data">
```

Cal remarcar el fet que s'ha hagut d'afegir el camp "enctype="multipart/form-data" per a poder gestionar l'enviament tant de dades de formulari com de fitxers adjunts.

A més, és necessari configurar el tamany màxim dels fitxers de vídeo adjunts que accepta el servidor al document *php.ini* del mòdul PHP. Els camps a modificar són els següents:

- *post_max_size*: limita el tamany de dades d'una operació tipus "POST". Per defecte, el seu valor és de 8MB. El modifiquem a 510MB.
- *upload_max_filesize*: limita el tamany del fitxer que es vol pujar al servidor. Per defecte, el seu valor és de 2MB. El modifiquem a 500MB.

També hem de configurar el directori on es guarda temporalment el fitxer durant un procés d'*upload* al servidor:

```
upload_tmp_dir = "C:/Program Files (x86)/WampServer/tmp"
```

L'aplicació retorna error, i en conseqüència no afegeix la jugada, en cas que el format del fitxer de vídeo no sigui *mp4* i/o en cas que s'excedeixi el tamany màxim del fitxer.

És important destacar que el *codec* del vídeo *mp4* ha de ser H.264 i que la descàrrega progressiva, que permet començar a visualitzar el vídeo abans que finalitzi la seva descàrrega, ha d'estar habilitada. Això implica que el *moov atom* [32] de la capçalera ha d'estar a l'inici del fitxer, ja que és necessari per a poder començar la reproducció. Si aquest àtom es troba al final, el client hauria de descarregar la totalitat de les dades del vídeo abans de poder accedir-hi.

Per a comprovar la posició del *moov atom* es pot utilitzar el programa *AtomicParsley* [32, 33] executant la següent sentència:

```
AtomicParsley.exe video.mp4 -T
```

A l'annex B es mostra l'anàlisi de dos vídeos: el primer no té habilitada la descàrrega progressiva; el segon sí.

Si tenim un vídeo *mp4* H.264 sense la descàrrega progressiva habilitada, podem moure el *moov atom* amb el programa *qt-faststart* [34] executant la següent sentència:

```
qt-faststart.exe video.mp4
```

Tornant a les funcionalitats del *webportal*, per a generar les imatges JPG i GIF corresponents a cada un dels vídeos de les noves jugades, s'han generat dos *scripts batch* que es criden directament des del codi PHP del document d'inserció d'una nova jugada.

Per a generar la imatge JPG s'ha utilitzat *ffmpeg* [35], un programa disponible en Windows i Linux que permet, entre moltes altres coses, extreure un *frame* específic d'un vídeo *mp4*. La sentència a executar és la següent:

```
ffmpeg.exe -i "video_input.mp4" -ss 2 -f image2 -s 320x180  
"imatge_output.jpg"
```

Amb el paràmetre “-ss” especifiquem quan volem extreure el *frame* (començant des de l'inici i comptant en segons). El paràmetre “-s” ens permet indicar la resolució que volem que tingui la imatge generada.

La nostra intenció era generar també la imatge GIF amb *ffmpeg*. Ho hem provat executant la següent sentència:

```
ffmpeg.exe -i ."video_input.mp4" -ss 1 -t 5 -r 2 -s 320x180 -  
pix_fmt rgb24 "imatge_output.gif"
```

on el paràmetre “-t” indica els segons que ha de durar la captura del vídeo, i “-r” fa referència al *rate de frames* per segon del GIF animat resultant. El GIF animat es genera correctament, però amb una qualitat inacceptablement baixa degut a que no s'utilitzen la totalitat de 256 tons dels colors primaris corresponents al format de píxel *rgb24* (l'únic que accepta *ffmpeg* per a generar GIF animats).

Degut a aquest inconvenient, s'ha optat per utilitzar un altre *software* executable per línia de comandes per a generar els GIF animats: *ImageMagick* [36]. El procediment consta de dues parts:

a) Extracció dels *frames* que formaran en GIF animat amb *ffmpeg*:

```
ffmpeg.exe -i "video_input" -ss 5 -t 15 -r 5 -s 320x180  
"img-%%d.gif"
```

Amb aquesta sentència es generen un conjunt d'imatges estàtiques GIF numerades gràcies a la nomenclatura "img-%%d.gif".

b) Animació del GIF a partir dels *frames* utilitzant *ImageMagick*:

```
convert -delay 20 -loop 0 img*.gif "imatge_output.gif"
```

El paràmetre "-delay" permet especificar la separació temporal entre dos *frames* consecutius. El paràmetre "-loop 0" indica que l'animació s'ha de repetir contínuament.

Tant les imatges JPG com les GIF es dipositen al servidor amb els seus noms corresponents en funció del número de jugada.

El diagrama de la Fig 3.11 mostra el procediment complet de l'actualització de la base de dades mitjançant el *webportal*.

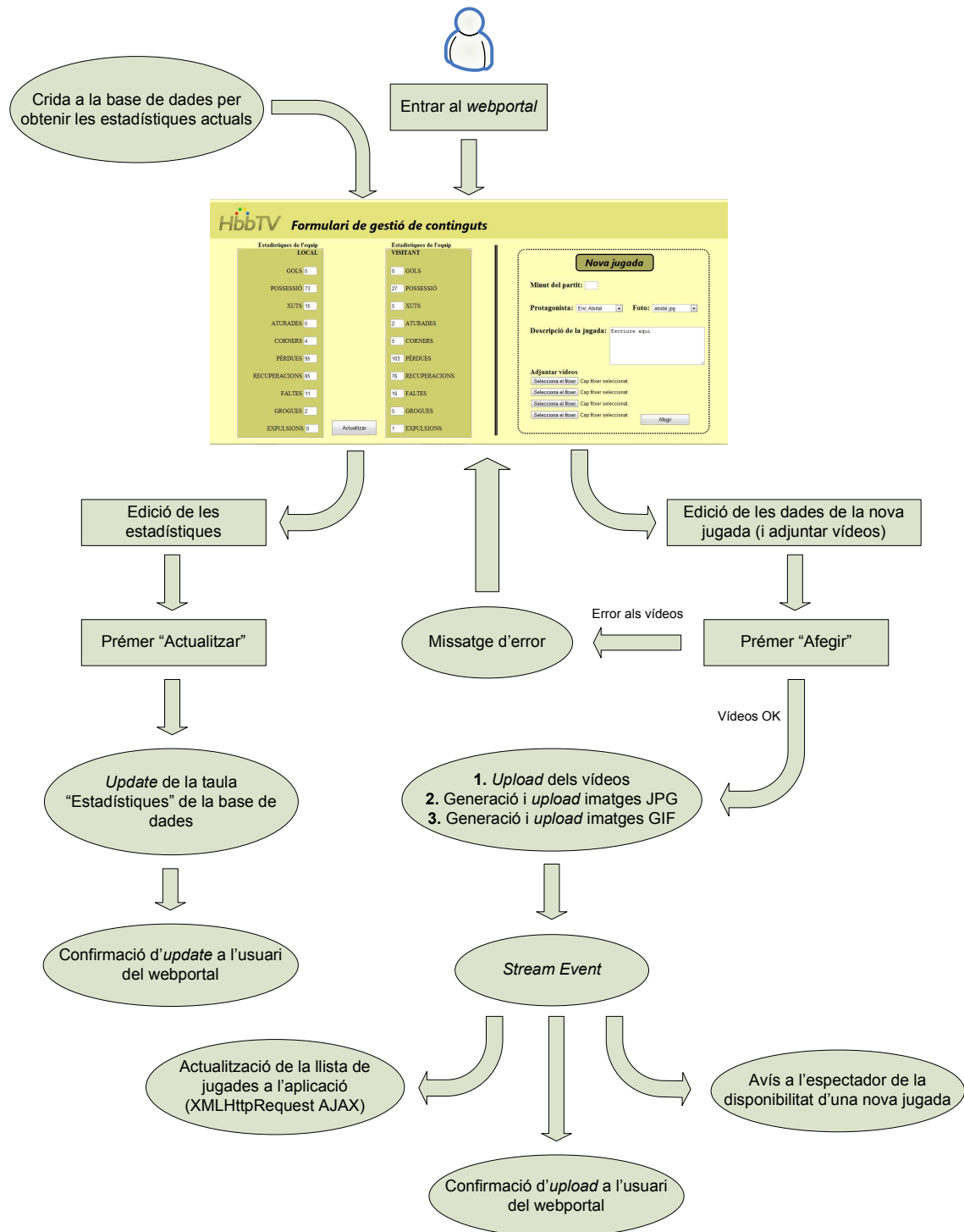


Fig. 3.11 Diagrama d'actualització de la base de dades mitjançant el *webportal*

CAPÍTOL 4. PROVES I RESULTATS

Un cop implementada l'aplicació amb l'ajuda dels entorns d'execució ja comentats anteriorment, s'ha procedit a realitzar la simulació del cicle *broadcast* complet: generació d'un *Transport Stream* amb l'aplicació senyalitzada, emissió en RF mitjançant una tarja moduladora DVB-T i recepció del senyal en dispositius compatibles amb l'estàndard HbbTV⁷.

Per a generar el TS s'ha utilitzat una eina software gratuïta: *Opencaster* [37, 38-40]. Un cop generat, s'ha d'emetre en RF per a que pugui ser visualitzat a través del *set-top box* d'Engel o la televisió connectada Samsung SmartTV, els dos dispositius HbbTV dels quals disposem. Per a assolir aquest objectiu s'ha utilitzat la tarja moduladora DTA-111 de Dektec (annex C) i el programa *StreamXpress* (annex D), que ens permet configurar els paràmetres de la modulació. A més, amb el software *TSReader Lite* [41] podem visualitzar l'estructura del *Transport Stream* generat.

4.1 Generació del *Transport Stream*

Opencaster està format per un conjunt d'eines desenvolupades per l'empresa italiana Avalpa que permeten generar continguts encapsulats en MPEG-2 *Transport Stream*. Aquestes eines funcionen en sistemes operatius Linux Debian i s'executen a través del terminal al no disposar d'interfície gràfica. El TS generat es pot emmagatzemar en un fitxer, pot ser enviat a través d'una xarxa IP o, com en el nostre cas, es pot modular amb una tarja DVB-T per a emetre'l en RF.

Opencaster ofereix una fàcil adaptació dels continguts als requisits DVB-T, permetent multiplexar els TS de vídeo i àudio amb les taules PSI-SI corresponents. A més, ens permet generar les pròpies taules mitjançant l'execució de *scripts* en llenguatge Python. Als tutorials del programa s'ofereixen exemples de *scripts* Python, de manera que tan sols és necessari realitzar petits canvis per a adaptar la informació de les taules a les nostres necessitats. A l'executar l'*script* que es troba a l'annex E obtenim els següents fitxers:

pat.ts	nit.ts	sdt.ts	pmt.ts	ait.ts
--------	--------	--------	--------	--------

Finalment només cal multiplexar aquestes taules amb els TS de vídeo i àudio. Això es fa amb l'eina *tscbrmuxer* d'*Opencaster*.

Així doncs, el procediment de generació d'un TS apte per a modular i emetre es divideix en les següents accions diferenciades:

⁷ Degut a que no s'ha disposat de la tarja moduladora fins dues setmanes abans de tancar la memòria del TFC, no s'ha pogut provar la senyalització de *Stream Events* en un TS modulad, però sí s'ha pogut emular amb el FireHbbTV.

- Configurar l'*script* de Python i executar-lo per a generar les taules PSI-SI. Per a fer-ho ens situem al directori on es troba l'*script* i executem la següent sentència:

```
root@user-desktop:/path# ./hbbtv-http.py
```

on "hbbtv-http.py" és el nom de l'*script*. La seva configuració es mostra en l'annex E.

- Multiplexar vídeo, àudio i taules PSI-SI executant una sentència del tipus:

```
root@user-desktop:/path# tscbrmuxer b:2300000 video.ts
b:188000 audio.ts b:3008 pat.ts b:3008 pmt.ts b:1500 sdt.ts
b:1400 nit.ts b:2000 ait.ts b:10772084 null.ts > ts_out.ts
```

Es pot veure que a la sentència s'indiquen tots els fitxers a multiplexar juntament amb el seu corresponent *bitrate*. El *bitrate* del TS resultant (ts_out.ts) serà la suma del conjunt. Observem que també afegim un fitxer anomenat "null.ts" (proporcionat per *Opencaster*). Això es fa per a que el *bitrate* resultant encaixi amb el d'un dels casos definits per l'estàndard DVB-T, que depenen de la modulació, el CR (*convolutional rate*) i l'interval de guarda. Els *bitrates* d'un TS modulats per DVB-T estan tabulats en funció d'aquests tres paràmetres.

El *bitrate* assignat a cada una de les taules PSI-SI depèn d'una banda de la seva mida i de l'altra de la freqüència amb què aquestes han de ser enviades. La PAT, per exemple, s'ha de refrescar cada 500 ms com a mínim. En funció del seu contingut, la mida de la taula oscil·la entre el d'un únic paquet TS (188 bytes = 1504 bits) i 23 paquets TS. En el nostre cas totes les taules ocupen tan sols un paquet TS, de manera que, seguint amb l'exemple de la PAT, en un segon haurem d'enviar 2 paquets i, per tant, 3008 bits. La resta de taules (exceptuant la PMT) s'envien amb menys freqüència, fet que implica un menor *bitrate* mínim.

No hi ha cap inconvenient en assignar un major *bitrate* a les taules PSI-SI, però aleshores caldrà tenir en compte que s'haurà de rebaixar el del fitxer "null.ts" per a que el total segueixi coincidint amb el tabulat per l'estàndard DVB-T.

Un cop tenim el fitxer *.ts, el podem analitzar amb el programa *TSReader Lite*. Si tenim el fitxer emmagatzemat a l'ordinador, hem de seleccionar *input source = file.dll* i obrir-lo. Si es disposa d'una tarja receptora DVB-T amb entrada d'antena també hi ha la possibilitat de connectar la sortida del modulador al PC per analitzar el TS emès. En aquest cas haurem de seleccionar l'*input source* que correspongui.

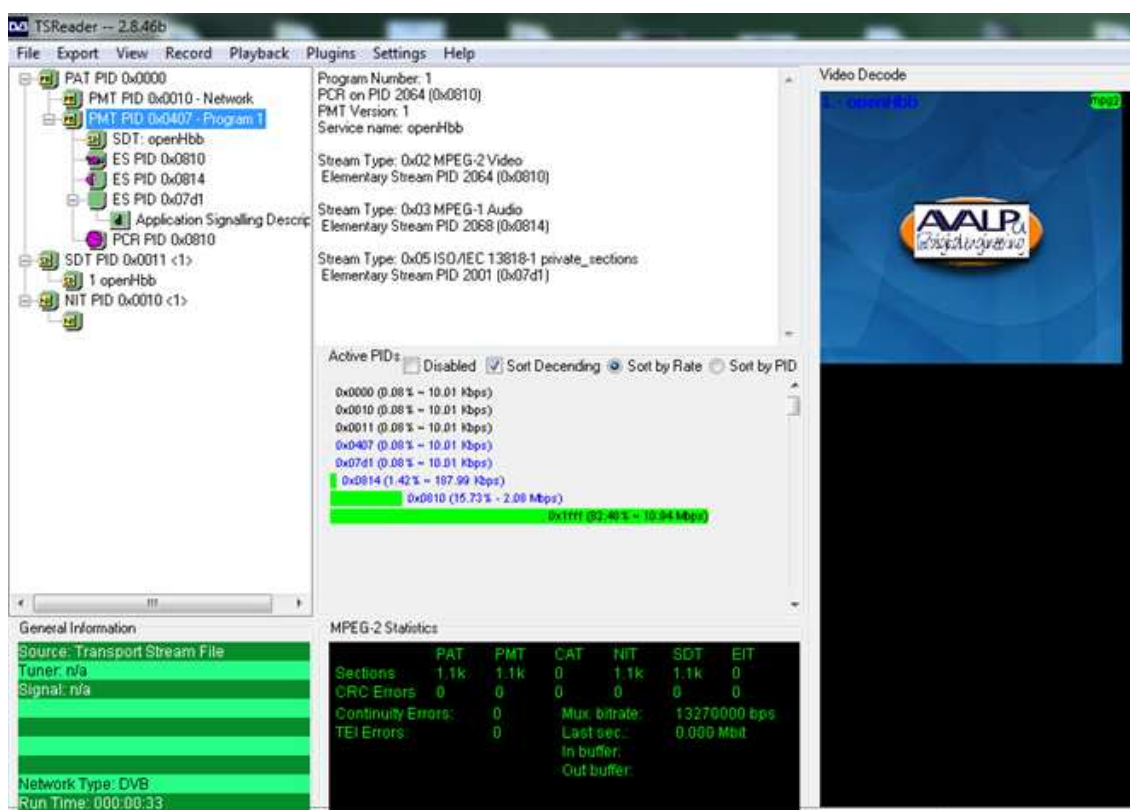


Fig. 4.1 Captura de pantalla de la interfície gràfica del programa TSReader

Veiem que es poden identificar fàcilment les diferents taules PSI-SI generades, així com els seus corresponents PIDs. En aquest cas es mostra seleccionada la PMT corresponent a l'únic programa o servei que hi ha al TS. A la part esquerra es mostra informació de la taula seleccionada. Podem observar que aquest servei consta de tres *streams*: *MPEG-2 Video*, *MPEG-1 Audio* i *ISO/IEC 13818-1 private_sections*. Aquest últim *stream* és el corresponent a la taula AIT on es fa la senyalització de l'aplicació HbbTV. És una secció privada i, per tant, no se'n pot visualitzar la informació amb el *TSReader Lite*.

4.2 Modulació i emissió del Transport Stream

El DTA-111 de Dektec (Fig 4.2) és un modulador multiestàndard que a l'hora permet convertir el senyal en banda base a l'entrada a VHF/UHF a la sortida. Té una sortida RF que permet una connexió directa amb l'entrada d'antena del dispositiu receptor.



Fig. 4.2 Modulador DTA-111 de Dektec

Per a emetre el *Transport Stream* a través de la sortida RF del modulador cal instal·lar els *drivers* de la tarja i un *software* que permeti configurar els paràmetres de la modulació i activar o desactivar l'emissió. Ens hem decantat pel *DTC-300 StreamXpress*, un *software* de Dektec per a màquines Windows que suporta qualsevol adaptador Dektec a la sortida, inclòs el DTA-111.

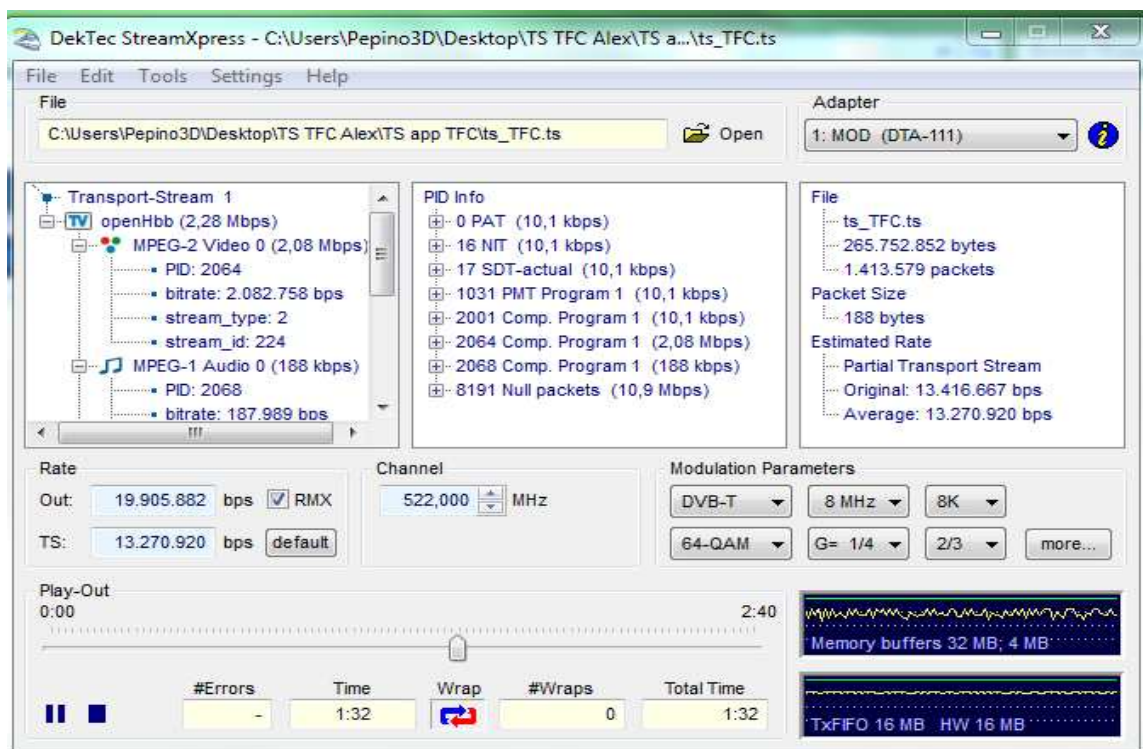


Fig. 4.3 Captura de pantalla de la interfície gràfica del programa *StreamXpress*

A la Fig 4.3 es pot observar que la interacció amb aquest programa és senzilla: primer cal seleccionar el TS que es vol emetre i l'adaptador on es troba el modulador. Com que només tenim una tarja moduladora, el MOD (DTA-111) ja

apareix per defecte. De la mateixa manera que el *TSReader Lite*, també se'n ofereix informació sobre les taules i els diferents *streams*, tot i que en un menor grau de detall.

Els paràmetres de la modulació són els corresponents a les emissions sobre l'estàndard DVB-T a l'Estat espanyol. Amb aquesta configuració, el *bitrate* de sortida queda perfectament definit: 19.905.882 bps. El *bitrate* del TS a l'entrada és de 13.270.920 bps perquè estava preparat per una modulació 16-QAM. Aquesta diferència no és un inconvenient, ja que el propi *StreamXpress* s'encarrega d'afegir paquets (*null packets*) per a adaptar el *rate* de sortida a la modulació seleccionada.

Com a alternativa, si es vol instal·lar la tarja moduladora en una màquina Linux per a emetre des d'allà, existeix un programa anomenat *DtPlay* proporcionat també per Dektec. No disposa d'interfície gràfica sinó que, de la mateixa manera que l'*Opencaster*, s'executa a través de línies de comandes. La sintaxi de les sentències a executar és la següent:

```
DtPlay playfile -r Rate [-m Mode][-t Type][-n Number][-p
ON/OFF][-mf Frequency][-mt ModType][-mc CodeRate][-f
YES/NO] [-s][-?]
```

A [42] s'ofereix la informació detallada de cada un dels camps de configuració.

4.3 Recepció amb el *set-top box* EN2000 d'Engel

L'escenari del muntatge per a fer les proves amb el receptor Engel és el que és mostra a la Fig 4.4.

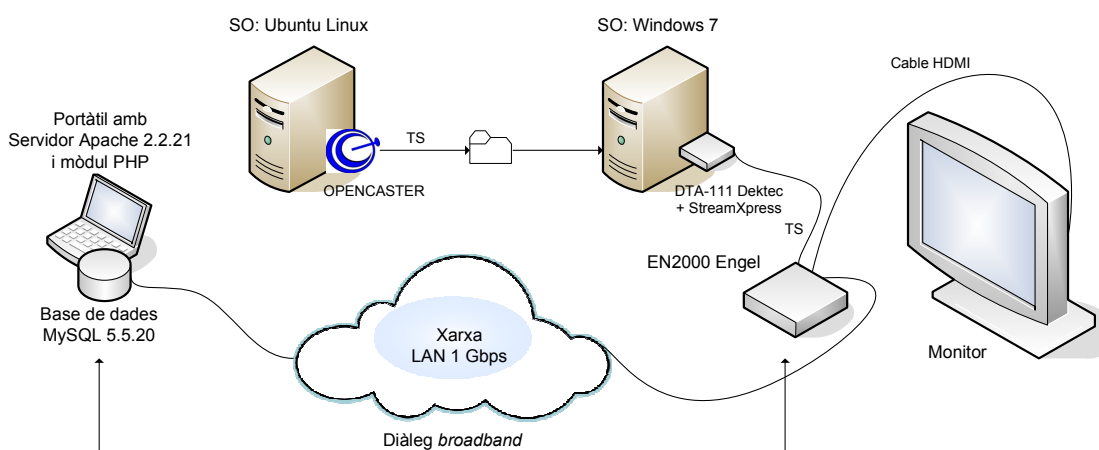


Fig. 4.4 Escenari de proves amb el receptor EN2000 d'Engel

L'aplicació, senyalitzada en mode *AUTOSTART*, es llança correctament al sintonitzar el canal. A la pàgina principal detectem alguns problemes:

- Estils CSS: no s'apliquen els arrodoniments inserits als *borders* d'alguns elements. Tampoc es visualitza l'efecte de *shadowing* o ombrejat sobre l'element que té el focus.
- *Widget* de Twitter: no apareix.
- Marges: alguns elements propers als marges exteriors queden retallats pel fet de que s'ha produït una lleugera disminució de la zona de visualització efectiva.
- Colors: l'aplicació pateix un cert grau de degradació visual degut a la menor disponibilitat de colors.

Tots aquests problemes afecten exclusivament a l'aparença gràfica de l'aplicació. No s'ha detectat cap problema de funcionalitat: podem navegar correctament per les diferents opcions del menú i es mostren les dades adequades en cada moment.

Si seleccionem una jugada es carrega la totalitat de la informació relacionada sense problemes. Es poden visualitzar els vídeos. A diferència de FireHbbTV, el receptor Engel sí que ens permet fer un *fast-forward* del vídeo sota demanda.

També s'ha comprovat que el navegador accepta *cookies*, ja que un cop s'ha emès el vot no es permet tornar a votar fins que aquestes caduquen.

Vist el comportament de l'aplicació executada des del receptor Engel s'ha optat per fer lleugeres modificacions d'estils per a que la compatibilitat sigui perfecta:

- Eliminació dels estils CSS d'arrodoniment i *shadowing*
- Eliminació del widget de Twitter
- Lleugera redistribució dels elements per a respectar els marges de seguretat evitant que aquests es retallin parcialment

L'aparença de l'aplicació personalitzada per al receptor Engel es pot veure a l'annex H. Cal destacar que aquesta nova versió és perfectament compatible amb FireHbbTV.

4.4 Recepció amb la televisió Samsung SmartTV

L'escenari del muntatge per a fer les proves amb la televisió connectada Samsung SmartTV és el que es representa a la Fig 4.5.

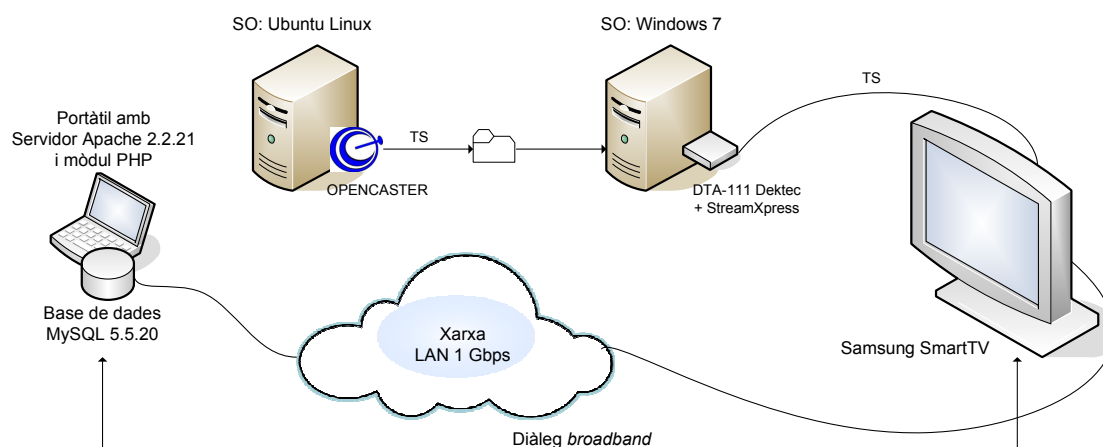


Fig. 4.5 Escenari de proves amb la televisió Samsung SmartTV

L'aplicació, senyalitzada en mode *AUTOSTART*, no es llança al sintonitzar el canal. Al no disposar de cap eina d'execució pas a pas per detectar possibles errors, s'ha optat per generar nous *Transport Streams* amb la senyalització de versions antigues de la mateixa aplicació per comprovar a partir de quina d'elles deixa de llançar-se amb normalitat. En aquest punt, s'ha fet una comparació manual entre el codi de dues versions consecutives: la que no funciona i la immediatament anterior que sí que funciona.

S'ha provat de fer canvis al codi de la versió més nova amb l'objectiu d'eliminar les noves funcionalitats implementades respecte l'anterior. Cap d'aquests canvis ha tingut èxit. Finalment, després de moltes revisions al codi, s'ha localitzat la font de l'error:

```

window.onload = function(){
    registerKeyEventListener();
    initApp();
    BCASOn();
    runStreamEvents('dvburl');
    updateJugades();
    runStep('bcast_pos_inici');
    document.getElementById("bot1").focus();
};

```

Aquest és el codi JavaScript que s'executa un cop s'han carregat la totalitat de fitxers de l'aplicació. Hi ha una diferència entre el mateix tros de codi de les dues versions de l'aplicació: en la més antiga (la que sí que funciona) la sentència "`document.getElementById("bot1").focus();`" es troba a la primera línia. En la versió més nova aquesta mateixa sentència està a l'última línia, tal com es pot observar.

Sorprenentment, al moure aquesta línia al principi del codi, la nova versió de l'aplicació es llança correctament. Aplicant el mateix canvi a la versió més actual també se soluciona el problema. L'explicació a aquest fenomen no és trivial, i la podríem trobar en el fet que el navegador de la Smart TV es troba tot

just en fase de desenvolupament i millora, de manera que es poden produir accions i/o problemes inesperats.

Un cop visualitzem l'aplicació a la Samsung SmartTV detectem alguns problemes de compatibilitat:

- Estils CSS: no s'apliquen els arrodoniments inserits als *borders* d'alguns elements. Tampoc es visualitza l'efecte de *shadowing* o ombrejat sobre l'element que té el focus.
- *Widget* de Twitter: no apareix.
- Imatges GIF: no apareixen.

Com que en el cas del receptor Engel EN2000 els dos primers problemes s'han solucionat creant una versió adaptada de l'aplicació, utilitzem aquesta nova versió per a continuar les proves amb la televisió Samsung SmartTV, canviant les imatges GIF per d'altres PNG.

A diferència de les proves amb el receptor Engel, amb la televisió Samsung detectem problemes de navegació per l'aplicació: ens podem moure correctament pels quatre botons del menú principal però no podem navegar per les jugades ni seleccionar-ne cap. Fent noves revisions i proves amb el codi JavaScript s'ha comprovat que el navegador de la Samsung SmartTV no comprèn la propietat `activeElement`. Així doncs, la sentència `document.activeElement` no retorna l'element que té el focus. Tal com està dissenyat i estructurat el codi és absolutament essencial trobar una alternativa semblant per a poder detectar on està el focus en cada instant i aplicar les accions necessàries. La solució l'hem trobat a [43]:

```
function _dom_trackActiveElement(evt) {
  if (evt && evt.target) {
    document.activeElement =
      evt.target == document ? null : evt.target;
  }
}

function _dom_trackActiveElementLost(evt) {
  document.activeElement = null;
}
```

Afegint aquestes dues funcions al codi JavaScript podem controlar en tot moment on és el focus executant el següent codi cada cop que es produeix un event del tipus *keydown*:

```
if (document.addEventListener) {
  document.addEventListener("focus", _dom_trackActiveElement, true);
  document.addEventListener("blur", _dom_trackActiveElementLost, true);
}
```

Un cop superat aquest obstacle ja podem accedir a les diferents jugades de la llista. Tot i així, sorgeix un nou problema: al desplaçar-nos per les diferents jugades, si premem el botó *VK_UP* sortim de la llista i el focus se situa al botó “Estadístiques”. Això es produeix degut a que s’executa l’event associat per defecte a aquest botó sense fer cas al mètode `preventDefault()` que s’hauria d’executar al prémer *VK_UP*. Davant la impossibilitat de trobar una alternativa adequada a aquest mètode, s’ha optat per assignar tecles diferents per a navegar per les jugades: enlloc d’utilitzar *VK_UP* i *VK_DOWN* s’utilitzarà *VK_LEFT* i *VK_RIGHT*, que no tenen cap acció assignada per defecte al produir-se un event del tipus *keydown*. S’han afegit unes imatges d’ajuda a la navegació a l’antiga posició del *widget* de Twitter per a facilitar la interacció de l’usuari amb l’aplicació.

Finalment, també s’ha hagut de fer una reordenació de les línies de codi que s’executen al prémer el botó *stop* per aturar la reproducció d’un vídeo, degut a que les diferents accions no s’executaven correctament. Es tracta d’un problema de la mateixa tipologia que el primer que s’ha comentat.

Destacar també que s’ha comprovat que el navegador no accepta *cookies*, ja que un cop s’ha emès el vot tenim l’opció de tornar a votar immediatament.

L’aparença de l’aplicació personalitzada per a la televisió connectada Samsung SmartTV es pot veure a l’annex H. Cal destacar que aquesta nova versió és perfectament compatible amb el receptor Engel EN2000 i amb FireHbbTV, és a dir, amb tots els dispositius que s’han pogut provar.

CONCLUSIONS I LÍNIES FUTURES

5.1 Conclusions

L'objectiu principal del treball s'ha assolit: disposem d'una aplicació HbbTV senyalitzada en un *Transport Stream* apte per a ser executada des de qualsevol dispositiu compatible amb l'estàndard. A més, també s'ha desenvolupat una eina mitjançant la qual podem gestionar els continguts de la base de dades que han d'aparèixer a l'aplicació.

Des del punt de vista de la funcionalitat s'ha aconseguit implementar des de zero una aplicació HbbTV seguint les especificacions marcades per la primera versió de l'estàndard aprovada per la ETSI. Cal destacar que actualment la disponibilitat d'exemples d'aplicacions comercials on poder contrastar dubtes és molt limitada. Hem aconseguit implementar una aplicació multidispositiu (emuladors, *set-top box*, Smart TV) que es comporta igual en les diferents plataformes d'execució. És necessari remarcar que no s'ha pogut implementar una aplicació amb la funcionalitat de Twitter per al *set-top box* i la Smart TV, però sí per als emuladors HbbTV disponibles (FireHbbTV i Opera TV Emulator). Pel que fa al disseny de grafisme amb CSS, als dispositius reals hem trobat limitacions amb els estils d'arrodoniment de *borders* i d'ombrejat. Aquestes limitacions no apareixen als emuladors. En aquest sentit, entenem que aquest no és un problema del nostre disseny, sinó més aviat de la implementació del *firmware* per part dels fabricants.

S'ha provat l'aplicació sobre equips reals amb el modulador DTA-111 de Dektec i dos receptors HbbTV: el *set-top box* EN2000 d'Engel i la Samsung SmartTV. Amb *Opencaster* hem pogut generar les taules de senyalització d'un *Transport Stream* i multiplexar-les posteriorment amb els *streams* de vídeo i àudio. En *Transport Stream* obtingut és analitzable amb el *TSReader Lite*. Per a emetre'l a través de la interfície de RF de la tarja hem configurat els paràmetres de la modulació mitjançant *StreamXpress*, que també ens permet acitvar i desactivar l'emissió.

Un cop dit això, cal remarcar que queden una gran quantitat de punts oberts, molts dels quals no entraven en la planificació inicial d'aquest TFC.

5.2 Línies futures

El primer punt susceptible de millora és el de donar una major fluïdesa al conjunt del desenvolupament. Idealment la transmissió del TS s'hauria de fer des de la mateixa màquina on aquest es genera. Això es podria fer configurant correctament els paràmetres de la modulació DVB amb el *software DtPlay* de Dektec. *Opencaster* proporciona eines per a generar un TS "obert" del tipus FIFO (*First In First Out*). No és un fitxer tancat i fix, sinó que es tracta d'un TS en el qual entren dades constantment (i s'eliminen les més antigues per a evitar una mida excessiva). Això permetria fer canvis a la senyalització en temps real.

De la mateixa manera, podríem enviar *Stream Events* en el moment oportú per avisar de la disponibilitat d'una nova jugada.

Encara es podria anar més enllà: hi hauria la possibilitat de relacionar l'edició d'una nova jugada des del *webportal* amb la generació d'un *Stream Event* amb *Opencaster*. Caldria implementar un *script batch* de generació de *Stream Events* que s'executés a l'afegir una nova jugada.

Pel què fa al *webportal*, es podria completar per a que no fos necessària l'eina d'administració de bases de dades *phpMyAdmin*. Actualment a través del *webportal* tan sols podem efectuar les accions més importants: editar les estadístiques i afegir una nova jugada. Si volem eliminar una jugada, editar-la o afegir un nou jugador a la llista ho hem de fer des de *phpMyAdmin*.

Respecte a l'aplicació HbbTV en sí, hi ha múltiples vies per a continuar el desenvolupament:

- Ampliació de la funcionalitat de Twitter: tal com es descriu a l'apartat 3.3.1.2 del treball, seria interessant poder interactuar amb Twitter directament des del dispositiu HbbTV sense necessitat d'accedir a un ordinador, una *tablet* o un *smartphone*.
- Ampliació de les funcionalitats de l'aplicació a partir de les millores afegides a la versió 1.5 de l'estàndard HbbTV: reproducció de continguts MPEG-DASH, ús de tecnologies DRM, etc.
- Inclusió de publicitat a l'aplicació gestionada per un *AdServer*: es podria incloure publicitat de tipologia diversa: imatges, vídeos (*preroll*, *midroll*, *postroll*, etc).
- Interacció gestual amb l'aplicació: mitjançant la plataforma *Kinect* seria possible controlar l'aplicació amb gestos, sense necessitat de disposar d'un comandament a distància.
- Transició a una base de dades del tipus no relacional, com per exemple *MongoDB* [44], per assegurar una major capacitat d'escalabilitat.

Cal tenir present que els dispositius HbbTV encara es troben en fase de desenvolupament i millora de *firmware*. Tal com s'ha comprovat a les proves de funcionalitat del TFC, la televisió Samsung SmartTV i el *set-top box* EN2000 del fabricant Engel no es comporten de la mateixa manera a l'hora d'executar una mateixa aplicació. Segons experiències que hem tingut utilitzant televisions connectades d'altres fabricants (executant aplicacions diferents) aquest comportament no uniforme s'extén a tots els dispositius.

BIBLIOGRAFIA

- [1] – ETSI TS 102 796 v1.1.1 “*Hybrid Broadcast Broadband TV*”
- [2] – Carmen Vicente, “*Introducción a las Cabeceras Digitales*”, Barcelona, 2011
- [3] – Apunts de l’assignatura “Xarxes de Comunicacions” de l’EETAC (UPC), “Tema 3. Internet i xarxes IP”
- [4] – *HbbTV Specification Version 1.5*
- [5] – MPEG DASH Wikipedia: http://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP
- [6] – Consorci HbbTV: <http://www.hbbtv.org>
- [7] – CE-HTML Wikipedia: <http://en.wikipedia.org/wiki/CE-HTML>
- [8] – Open IPTV Forum – *Release 1 DAE Reference Guide v1.0*
- [9] – Open IPTV Forum – *Release 1 Specification volume 1 – Overview v1.1*
- [10] – Open IPTV Forum – *Release 1 Specification volume 2 – Media Formats v1.1*
- [11] – Open IPTV Forum – *Release 1 Specification volume 4 – Protocols v1.1*
- [12] – Open IPTV Forum – *Release 1 Specification volume 5 – Declaration Application Environment v1.1*
- [13] – ETSI TS 102 809 v1.1.1 “*Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in Hybrid broadcast / broadband environments*”
- [14] – Javier Eguíluz Pérez, “*Introducción a XHTML*”, <http://www.librosweb.es>
- [15] – *Recopilación de videotutoriales relacionados con la programación en PHP*: <http://www.desarrolloweb.com/manuales/videtutorial-php.html>
- [16] – *HTML tutorials. Cookies*: <http://www.yourhtmlsource.com/javascript/cookies.html>
- [17] – PHP manual: <http://es.php.net/manual/en/index.php>
- [18] – Javier Eguíluz Pérez, “*Introducción a CSS*”, <http://www.librosweb.es>
- [19] – Javier Eguíluz Pérez, “*Introducción a AJAX*”, <http://www.librosweb.es>

- [20] – Javier Eguíluz Pérez, “*Introducción a JavaScript*”,
<http://www.librosweb.es>
- [21] – Wikipèdia, Model Vista Controlador:
<http://en.wikipedia.org/wiki/Model%E2%80%93View%E2%80%93Controller>
- [22] – Jose Antonio Rodríguez, “*Tutorial de PHP y MySQL*”, 2000
- [23] – *WampServer, a Windows web development environment*:
<http://www.wampserver.com/en/>
- [24] – phpMyAdmin: http://www.phpmyadmin.net/home_page/index.php
- [25] – Complements Firefox. Fire HbbTV: <https://addons.mozilla.org/es-es/firefox/addon/firehbbtv/>
- [26] – Informació sobre Fire HbbTV: <http://tum-iptv.aw.atosorigin.com/firehbbtv/index.php>
- [27] – *Komodo Edit 7 for Python, PHP, Ruby, JavaScript, Perl and Web Dev*:
<http://www.activestate.com/komodo-edit>
- [28] – MIT-xperts HbbTV testsuite: <http://itv.mit-xperts.com/hbbtvtest/index.php>
- [29] – Carlos Eduardo Sotelo, “*Construcción de una aplicación web para crear un sistema de encuestas con la tecnología PHP y la base de datos MySQL*”:
<http://www.desarrolloweb.com/manuales/46/>
- [30] – Miguel Ángel Álvarez, “*Realizamos unas páginas PHP que reciben un archivo del disco duro del visitante y lo suben a un servidor remoto*”:
<http://www.desarrolloweb.com/articulos/1307.php>
- [31] – *PHP File Upload*: http://www.w3schools.com/php/php_file_upload.asp
- [32] – Maxim Levkov, “*Understanding the movie atom*”:
http://www.adobe.com/devnet/video/articles/mp4_movie_atom.html
- [33] – MP4 *moov atoms*. AtomicParlsey: <http://jordanjr.com/articles/drupal-7-mediafront-mpeg-4-moov-atoms-and-buffering>
- [34] – qt-faststart *download*: <http://multimedia.cx/eggs/improving-qt-faststart/>
- [35] – ffmpeg: <http://ffmpeg.org/>
- [36] – ImageMagick: <http://www.imagemagick.org/script/index.php>
- [37] – Opencaster 3.1: *the free digital TV software*: <http://www.avalpa.com/the-key-values/15-free-software/33-opencaster/>

- [38] – *Avalpa Broadcast Server User Manual*, release 3.0, 2011:
<http://www.avalpa.com/assets/freesoft/opencaster/AvalpaBroadcastServerUserManual-v3.0.pdf>
- [39] – Fernando Muñoz Fernández de Legaria, Pedro Ansorena González, “*Implementación de una solución para la gestión de un canal de televisión digital terrestre*”, Navarra, 2009
- [40] – Jonathan Ohana (hepia), “*Diploma work: Building of an HbbTV demonstrator*”, 2010
- [41] – TSReader: <http://www.tsreader.com/tsreader/index.html>
- [42] – DtPlay manual:
<http://www.bjpace.com.cn/product/dektec/manual/DtPlay%20Usage%20Instructions.pdf>
- [43] – Ivan Papelnjak, “*Emulating activeElement property with DOM browsers*”:
<http://ajaxandxml.blogspot.com.es/2007/11/emulating-activeelement-property-with.html>
- [44] – MongoDB: <http://www.mongodb.org/>

GLOSSARI

ADSL	<i>Asymmetric Digital Subscriber Line</i>
AIT	<i>Application Information Table</i>
API	<i>Application Programming Interface</i>
ARQ	<i>Automatic Repeat Request</i>
AVC	<i>Advanced Video Coding</i>
A/V	<i>Audiovisual</i>
CAT	<i>Conditional Access Table</i>
CE	<i>Consumer Electronics</i>
CEA	<i>Consumer Electronics Association</i>
CENELEC	<i>Comité Européen de Normalisation Electrotechnique</i>
DAE	<i>Declaration Application Environment</i>
DASH	<i>Dynamic Adaptive Streaming over HTTP</i>
DAVIC	<i>Digital Audio Video Coding</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DOM	<i>Document Object Model</i>
DRM	<i>Digital Rights Management</i>
DVB	<i>Digital Video Broadcasting</i>
EBU	<i>European Broadcasting Union</i>
ES	<i>Elementary Stream</i>
ETSI	<i>European Telecommunications Standards Institute</i>
GIF	<i>Graphics Interface Format</i>
HbbTV	<i>Hybrid Broadcast Broadband TV</i>
HD-SDI	<i>High Definition – Serial Digital Interface</i>
HEAAC	<i>High-Efficiency Advanced Audio Coding</i>
HTTP	<i>HyperText Transfer Protocol</i>
IEC	<i>International Electrotechnical Commission</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
ITU	<i>International Telecommunications Union</i>
JPEG	<i>Joint Photographic Experts Group</i>
MPEG	<i>Moving Picture Experts Group</i>
MySQL	<i>My Structured Query Language</i>
NIT	<i>Network Information Table</i>

NTSC	<i>National Television System Comitee</i>
OIPF	<i>Open IP TV Forum</i>
OSI	<i>Open System Interconnection</i>
PAL	<i>Phase Alternating Line</i>
PAT	<i>Program Association Table</i>
PES	<i>Program Elementary Stream</i>
PID	<i>Packet Identifier</i>
PMT	<i>Program Map Table</i>
PNG	<i>Portable Network Graphics</i>
POP3	<i>Post Office Protocol, versió 3</i>
PS	<i>Program Stream</i>
PSI/SI	<i>Program Specific Information / Service Information</i>
QAM	<i>Quadrature Amplitude Modulation</i>
RAM	<i>Random Access Memory</i>
RF	<i>Radiofrecuència</i>
RFC	<i>Request For Comments</i>
RTSP	<i>Real Time Streaming Protocol</i>
RTP	<i>Real Time Protocol</i>
SD-SDI	<i>Standard Definition – Serial Digital Interface</i>
SMPTE	<i>Society of Motion Picture and Television Engineers</i>
SSH	<i>Secure Shell</i>
TCP	<i>Transmission Control Protocol</i>
TDT	<i>Televisió Digital Terrestre</i>
TELNET	<i>Telecommunication Network</i>
TLS	<i>Transport Layer Security</i>
TS	<i>Transport Stream</i>
UDP	<i>User Datagram Protocol</i>

ANNEXOS

Annex A. Diagrames de comportement de les aplicacions

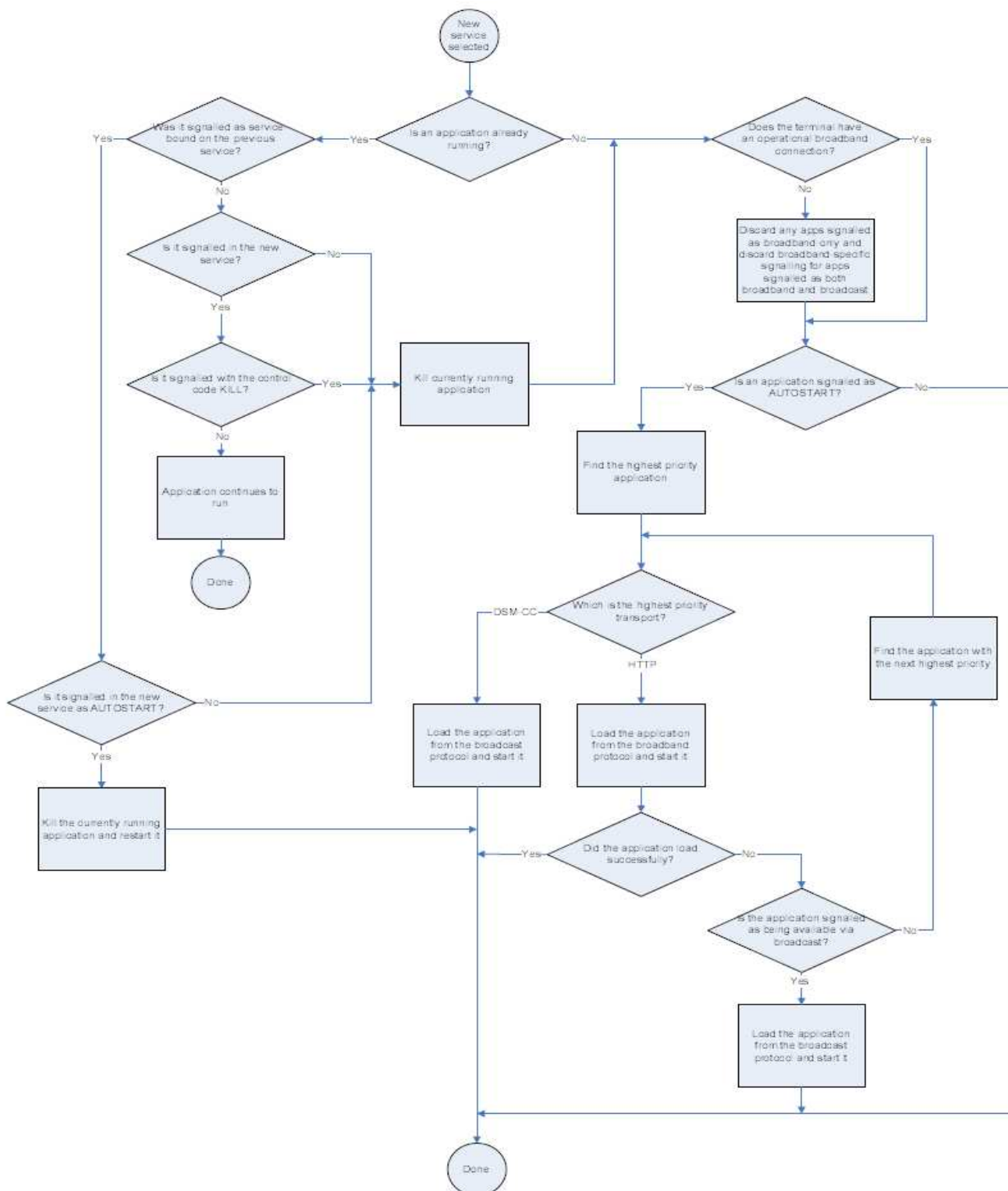


Fig. A.1 Comportament al seleccionar un canal *broadcast*

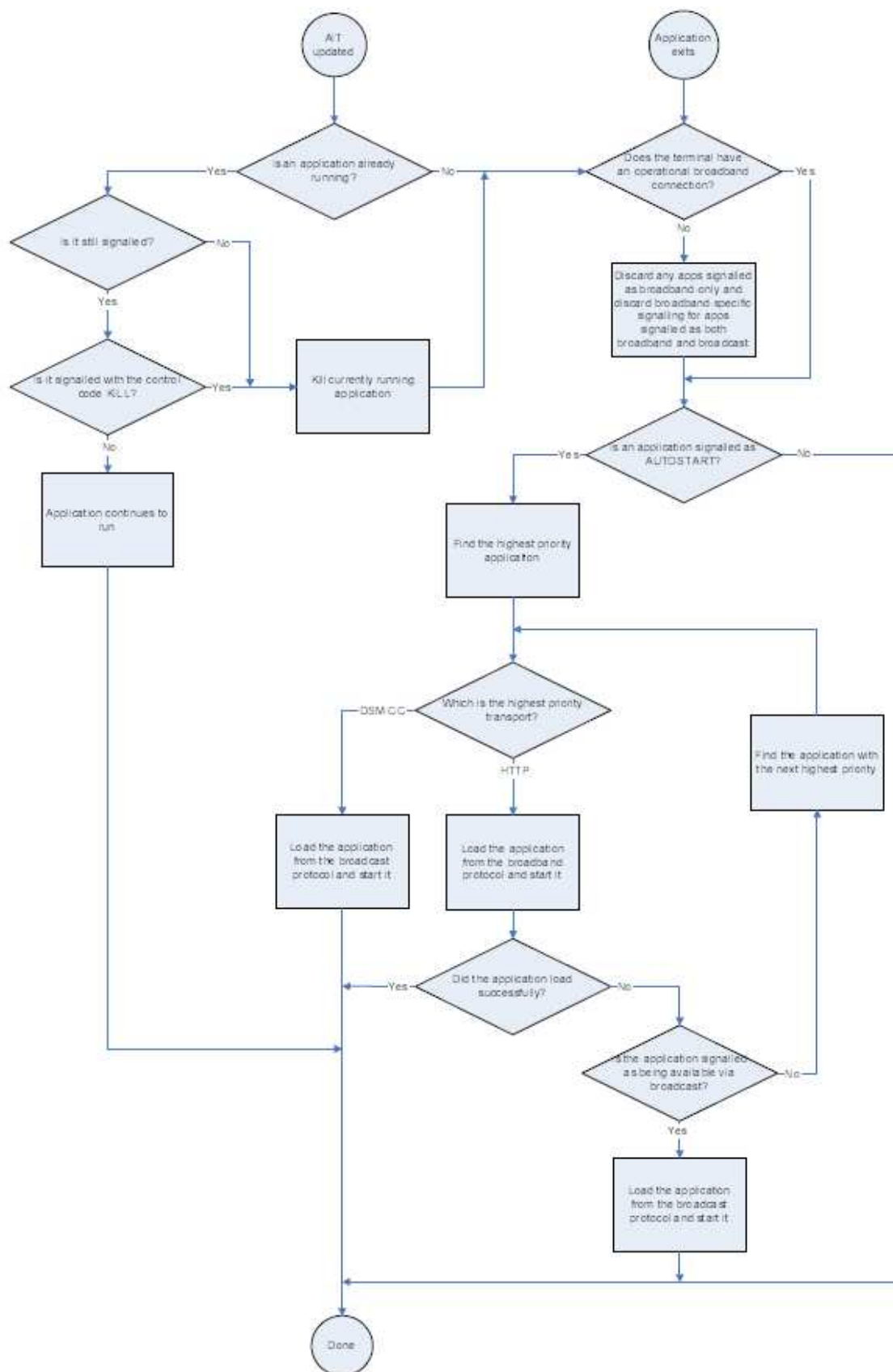


Fig. A.2 Comportament mentre un canal *broadcast* està seleccionat

A continuació, es mostren de forma esquemàtica les transicions entre diferents estats de varies aplicacions que tenen lloc davant de determinades accions de l'usuari. L'exemple està extret de l'estàndard HbbTV de la ETSI.

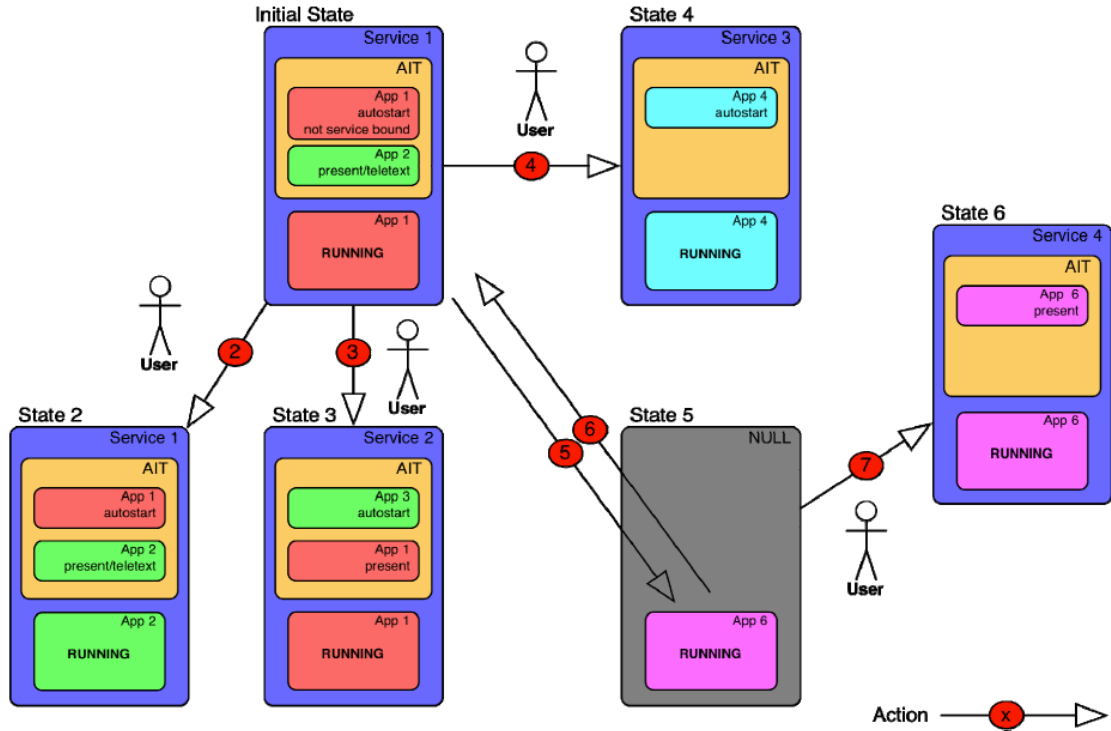


Fig. A.3 Cicle de vida de les aplicacions

Taula A.1. Explicació de les transicions entre estats

Estat inicial	Acció	Estat resultant
<i>Initial state:</i> s'està executant App 1	2: l'usuari prem el botó "TEXT"	<i>State 2:</i> S'inicia App 2 degut a la senyalització <i>TELETEXT</i>
<i>Initial state:</i> s'està executant App 1	3: l'usuari sintonitza el servei 2	<i>State 3:</i> App 1 es continua executant degut a la seva senyalització com a <i>PRESENT</i>
<i>Initial state:</i> s'està executant App 1	4: l'usuari sintonitza el servei 3	<i>State 4:</i> S'atura App 1 i s'inicia App 4 degut a la seva senyalització com a <i>AUTOSTART</i>
<i>Initial state:</i> s'està executant App 1	5: l'aplicació fa una crida <code>createApplication()</code> a una aplicació externa	<i>State 5:</i> S'executa l'aplicació independent App 6

<i>State 5:</i> s'està executant App 6	6: l'usuari sintonitza el servei 1	<i>State 1:</i> S'atura App 6 i s'inicia App 1 degut a la seva senyalització com a <i>AUTOSTART</i>
<i>State 5:</i> s'està executant App 6	7: l'usuari sintonitza el servei 4	<i>State 6:</i> App 6 es continua executant degut a la seva senyalització com a <i>PRESENT</i>

Annex B. Anàlisi estructural de fitxers MP4 amb *AtomicParsley*

Aquest primer anàlisi correspon a un vídeo sense la descàrrega progressiva habilitada. Cal observar que prèviament a descarregar el *moov atom* hem de descarregar el *mdat atom*, corresponent al conjunt de dades del vídeo complet.

```
Atom ftyp @ 0 of size: 32, ends @ 32
Atom free @ 32 of size: 8, ends @ 40
Atom mdat @ 40 of size: 8028289, ends @ 8028329
Atom moov @ 8028329 of size: 51551, ends @ 8079880
  Atom mvhd @ 8028337 of size: 108, ends @ 8028445
  Atom trak @ 8028445 of size: 15586, ends @ 8044031
    Atom tkhd @ 8028453 of size: 92, ends @ 8028545
    Atom edts @ 8028545 of size: 36, ends @ 8028581
      Atom elst @ 8028553 of size: 28, ends @ 8028581
    Atom mdia @ 8028581 of size: 15450, ends @ 8044031
      Atom mdhd @ 8028589 of size: 32, ends @ 8028621
      Atom hdlr @ 8028621 of size: 45, ends @ 8028666
      Atom minf @ 8028666 of size: 15365, ends @ 8044031
        Atom vmhd @ 8028674 of size: 20, ends @ 8028694
        Atom dinf @ 8028694 of size: 36, ends @ 8028730
          Atom dref @ 8028702 of size: 28, ends @ 8028730
        Atom stbl @ 8028730 of size: 15301, ends @ 8044031
          Atom stsd @ 8028738 of size: 173, ends @ 8028911
            Atom avc1 @ 8028754 of size: 157, ends @ 8028911
              Atom avcC @ 8028840 of size: 55, ends @ 8028895
              Atom pasp @ 8028895 of size: 16, ends @ 8028911
            ~
          Atom stts @ 8028911 of size: 24, ends @ 8028935
          Atom stss @ 8028935 of size: 104, ends @ 8029039
          Atom ctts @ 8029039 of size: 4920, ends @ 8033959
          Atom stsc @ 8033959 of size: 40, ends @ 8033999
          Atom stsz @ 8033999 of size: 5020, ends @ 8039019
          Atom stco @ 8039019 of size: 5012, ends @ 8044031
      Atom trak @ 8044031 of size: 35752, ends @ 8079783
        Atom tkhd @ 8044039 of size: 92, ends @ 8044131
        Atom edts @ 8044131 of size: 36, ends @ 8044167
          Atom elst @ 8044139 of size: 28, ends @ 8044167
        Atom mdia @ 8044167 of size: 35616, ends @ 8079783
          Atom mdhd @ 8044175 of size: 32, ends @ 8044207
          Atom hdlr @ 8044207 of size: 45, ends @ 8044252
          Atom minf @ 8044252 of size: 35531, ends @ 8079783
            Atom smhd @ 8044260 of size: 16, ends @ 8044276
            Atom dinf @ 8044276 of size: 36, ends @ 8044312
              Atom dref @ 8044284 of size: 28, ends @ 8044312
            Atom stbl @ 8044312 of size: 35471, ends @ 8079783
              Atom stsd @ 8044320 of size: 103, ends @ 8044423
                Atom mp4a @ 8044336 of size: 87, ends @ 8044423
                  Atom esds @ 8044372 of size: 51, ends @ 8044423
                Atom stts @ 8044423 of size: 13352, ends @ 8057775
                Atom stsc @ 8057775 of size: 8344, ends @ 8066119
                Atom stsz @ 8066119 of size: 8652, ends @ 8074771
                Atom stco @ 8074771 of size: 5012, ends @ 8079783
          Atom udta @ 8079783 of size: 97, ends @ 8079880
            Atom meta @ 8079791 of size: 89, ends @ 8079880
              Atom hdlr @ 8079803 of size: 33, ends @ 8079836
              Atom ilst @ 8079836 of size: 44, ends @ 8079880
                Atom @too @ 8079844 of size: 36, ends @ 8079880
                Atom data @ 8079852 of size: 28, ends @ 8079880
```

~ denotes an unknown atom

Total size: 8079880 bytes; 51 atoms total. AtomicParsley version: 0.9.0
(utf16)

En aquest segon cas, el *moov atom* es descarrega abans que el *mdat atom*, de manera que quan comença la descàrrega de dades ja es pot iniciar la reproducció del contingut.

```
Atom ftyp @ 0 of size: 32, ends @ 32
Atom moov @ 32 of size: 51551, ends @ 51583
  Atom mvhd @ 40 of size: 108, ends @ 148
  Atom trak @ 148 of size: 15586, ends @ 15734
    Atom tkhd @ 156 of size: 92, ends @ 248
    Atom edts @ 248 of size: 36, ends @ 284
      Atom elst @ 256 of size: 28, ends @ 284
    Atom mdia @ 284 of size: 15450, ends @ 15734
      Atom mdhd @ 292 of size: 32, ends @ 324
      Atom hdlr @ 324 of size: 45, ends @ 369
      Atom minf @ 369 of size: 15365, ends @ 15734
        Atom vmhd @ 377 of size: 20, ends @ 397
        Atom dinf @ 397 of size: 36, ends @ 433
          Atom dref @ 405 of size: 28, ends @ 433
        Atom stbl @ 433 of size: 15301, ends @ 15734
          Atom stsd @ 441 of size: 173, ends @ 614
            Atom avc1 @ 457 of size: 157, ends @ 614
              Atom avcC @ 543 of size: 55, ends @ 598
              Atom pasp @ 598 of size: 16, ends @ 614
            ~
          Atom stts @ 614 of size: 24, ends @ 638
          Atom stss @ 638 of size: 104, ends @ 742
          Atom ctts @ 742 of size: 4920, ends @ 5662
          Atom stsc @ 5662 of size: 40, ends @ 5702
          Atom stsz @ 5702 of size: 5020, ends @ 10722
          Atom stco @ 10722 of size: 5012, ends @ 15734
        Atom trak @ 15734 of size: 35752, ends @ 51486
          Atom tkhd @ 15742 of size: 92, ends @ 15834
          Atom edts @ 15834 of size: 36, ends @ 15870
            Atom elst @ 15842 of size: 28, ends @ 15870
          Atom mdia @ 15870 of size: 35616, ends @ 51486
            Atom mdhd @ 15878 of size: 32, ends @ 15910
            Atom hdlr @ 15910 of size: 45, ends @ 15955
            Atom minf @ 15955 of size: 35531, ends @ 51486
              Atom smhd @ 15963 of size: 16, ends @ 15979
              Atom dinf @ 15979 of size: 36, ends @ 16015
                Atom dref @ 15987 of size: 28, ends @ 16015
              Atom stbl @ 16015 of size: 35471, ends @ 51486
                Atom stsd @ 16023 of size: 103, ends @ 16126
                  Atom mp4a @ 16039 of size: 87, ends @ 16126
                    Atom esds @ 16075 of size: 51, ends @ 16126
                  Atom stts @ 16126 of size: 13352, ends @ 29478
                  Atom stsc @ 29478 of size: 8344, ends @ 37822
                  Atom stsz @ 37822 of size: 8652, ends @ 46474
                  Atom stco @ 46474 of size: 5012, ends @ 51486
            Atom udta @ 51486 of size: 97, ends @ 51583
              Atom meta @ 51494 of size: 89, ends @ 51583
                Atom hdlr @ 51506 of size: 33, ends @ 51539
                Atom ilst @ 51539 of size: 44, ends @ 51583
                  Atom @too @ 51547 of size: 36, ends @ 51583
                  Atom data @ 51555 of size: 28, ends @ 51583
          Atom free @ 51583 of size: 8, ends @ 51591
          Atom mdat @ 51591 of size: 8028289, ends @ 8079880

~ denotes an unknown atom
-----
Total size: 8079880 bytes; 51 atoms total. AtomicParsley version: 0.9.0
(utf16)
Media data: 8028289 bytes; 51591 bytes all other atoms (0.639% atom overhead).
-----
```

Annex C. Leaflet de la tarjeta moduladora DTA-111

DTA-111

DekTec

Multi-Standard Modulator with VHF/UHF Upconverter

- Low-cost multi-standard modulator
- Fixed output level
- Upgradable to support new standards

FEATURES

- Multi-standard modulator with support for most QAM-, OFDM- and VSB-based modulation standards
- Supports all constellations and modulation modes for each supported standard
- All-channel upconverter 47 .. 862MHz fully agile over VHF and UHF band
- Fixed output level of -15.0dBm for QAM and -18.0dBm for OFDM
- RF output for direct connection to the antenna input of a digital receiver
- Programming interface (DTAPI) is fully compatible with other DekTec digital-video output adapters
- PCI rev 2.2, 32 bit, 33 or 66MHz bus



APPLICATIONS

- General purpose test modulator
- SMATV, hotel server, shows and exhibitions
- OEM applications

KEY ATTRIBUTES

Parameter	Value
Frequency range	47 .. 862MHz \pm 1ppm
Bandwidth	5.0 .. 8.0MHz
MER (OFDM)	\approx 38dB (QAM) \approx 40dB (OFDM)
Connector	75- Ω F
Return loss	\geq 15dB (47 .. 862MHz)
Level (QAM)	-15dBm (+34dBmV)
Level (OFDM)	-18dBm (+31dBmV)
Phase noise	$<$ -90dBc @ 10kHz
Spectral purity	$>$ 50dB (47 .. 862MHz) $>$ 90dB (muted)
PCI compliancy	r2.2, 32-bit, 33/66MHz
OS	XP/2k3/Viasta/2k8/7 Linux 2.6

MODULATION STANDARDS

Modulation	Standard
ATSC VSB	ATSC A/53E
CMMB*	GY/T 220.1/2-2006
DTMB*	GB 20600-2006
DVB-C	EN 300 429
DVB-C2*	EN 302 769
DVB-T / DVB-H	EN 300 744
DVB-T2*	EN 302 755
ISDB-T*	ARIB STD-B31
QAM	J.83 Annex A/B/C

* Option

ORDERING INFORMATION

Type	Description
DTA-111-SP	VHF/UHF modulator with StreamXpress

Please refer to www.dektec.com for the latest pricing and a list of distributors and resellers.

Annex D. Leaflet del software DTC-300 StreamXpress

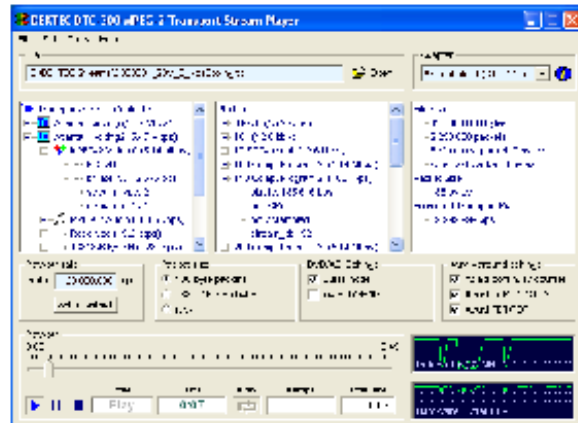
DTC-300 StreamXpress

Stream Player Software

- Integrated (P)SI viewer, looping
- Remote control option
- Selective error injection

FEATURES

- Integrate yourself a low-cost, high-performance stream player using a standard- or industrial PC, a DekTec output adapter and the StreamXpress playback software
- Integrated (P)SI viewer
- Control of modulation parameters
- Optional channel modeling of RF output signal
- Integrated ISDB-T hierarchical multiplexer
- Supports DTA-Plus RF attenuator
- Play-out at a higher rate by null-packet stuffing with PCR correction
- Endless play with optional automatic correction of continuity-counter and PCR/PTS/DTS fields
- Automatic computation of transport rate
- Reproducible injection of errors in the transport stream with adjustable error rate
- SOAP-based remote control option



APPLICATIONS

- Universal test-stream generator for feeding set-top boxes, digital TVs, digital-video processing equipment, etc.
- Demos at trade shows, in stores, etc.
- Test-signal generator for ATE environments

SUPPORTED OUTPUT ADAPTERS

Signal Type	Supported Adapters
DVB-ASI	DTA-100, DTA-112, DTA-115, DTA-140, DTE-3100, DTU-205
ASI/SDI	DTA-145, DTA-160, DTA-545, DTA-2136, DTA-2142, DTA-2144, DTA-2145, DTA-2152, DTA-2160, DTU-205, DTU-245
DVB-SPI	DTA-102, DTA-2142
Modulators	DTA-107S2, DTA-110T, DTA-111, DTA-112, DTA-115, DTA-116, DTA-117, DTA-2107, DTA-2111, DTU-215
TSoIP	DTA-160, DTA-2160
I/Q Samples	DTA-115, DTA-116, DTA-117, DTA-2107, DTA-2111, DTU-215

RELATED PRODUCTS

Type	Description
DTC-302-RC	Remote-control option
DTC-305-CM	Channel-modelling option

PC REQUIREMENTS

Platform	XP/2k3/Vista/2k8/Win7, 1GB
Processor*	PIII (ASI) .. Core 2 (DVB-T2)

* Or equivalent AMD processor

ORDERING INFORMATION

Type	Description
DTC-300	StreamXpress stream player software

Please refer to www.dektec.com for the latest pricing and a list of distributors and resellers.

Annex E. Cody Python d'Opencaster per a generar les taules PSI/SI

```
#!/usr/bin/env python

#
# Copyright 2010, mediatvcom (http://www.mediatvcom.com/), Claude
# Vanderm. Based on Lorenzo Pallara scripts (l.pallara@avalpa.com)
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
# USA

import os

from dvbobjects.PSI.PAT import *
from dvbobjects.PSI.NIT import *
from dvbobjects.PSI.SDT import *
from dvbobjects.PSI.PMT import *
from dvbobjects.DVB.Descriptors import *
from dvbobjects.MPEG.Descriptors import *
from dvbobjects.MHP.AIT import *
from dvbobjects.HBBTV.Descriptors import *

#
# Shared values
#

demo_transport_stream_id = 1 # demo value, an official value should be
demo_original_network_id = 1 # demo value, an official value should be
demo_service_id = 1 # demo value
pmt1_pid= 1031
ait1_pid = 2001
stel_pid = 2002

# parameters reported into the AIT to signalize a broadband
application.
appli_name = "my hbbtv application" #application name
appli_root = "http://my_application_root_path/myHbbTV-app/" #URL base
of transport_protocol_descriptor
appli_path = "index.html" #initial_path_bytes of simple application
descriptor. So the application path will be
"http://my_application_root_path/myHbbTV-app/index.html"
organisationId_1 = 10 # this is a demo value, dvb.org should assign
an unique value
```

```

applicationId_1 = 1001 # this is a demo value. This number corresponds
to a trusted application.
organisationId_2 = 11
applicationId_2 = 1002

# below, the video and audio PIDs reported into the PMT
video1_pid = 2064
audio1_pid = 2068
#
# Network Information Table
# this is a basic NIT with the minimum descriptors, OpenCaster has a
big library ready to use
#

nit = network_information_section(
    network_id = 1,
    network_descriptor_loop = [
        network_descriptor(network_name = "OpenHbb",),
    ],
    transport_stream_loop = [
        transport_stream_loop_item(
            transport_stream_id = demo_transport_stream_id,
            original_network_id = demo_original_network_id,
            transport_descriptor_loop = [
                service_list_descriptor(
                    dvb_service_descriptor_loop = [
                        service_descriptor_loop_item(
                            service_ID = demo_service_id,
                            service_type = 1, # digital tv service type
                        ),
                    ],
                ),
            ],
        ),
    ],
    version_number = 1, # you need to change the table number
every time you edit, so the decoder will compare its version with the
new one and update the table
    section_number = 0,
    last_section_number = 0,
)

#
# Program Association Table (ISO/IEC 13818-1 2.4.4.3)
#

pat = program_association_section(
    transport_stream_id = demo_transport_stream_id,
    program_loop = [
        program_loop_item(
            program_number = demo_service_id,
            PID = pmt1_pid,
        ),
        program_loop_item(
            program_number = 0, # special program for the NIT
            PID = 16,
        ),
    ],
    version_number = 1, # you need to change the table number
every time you edit, so the decoder will compare its version with the
new one and update the table
    section_number = 0,
)

```

```

        last_section_number = 0,
    )

#
# Program Map Table (ISO/IEC 13818-1 2.4.4.8)
# this is PMT for HbbTV interactive applications
#

pmt = program_map_section(
    program_number = demo_service_id,
    PCR_PID = video1_pid, # usually the same than the video
    program_info_descriptor_loop = [],
    stream_loop = [
        stream_loop_item(
            stream_type = 2, # mpeg2 video stream type
            elementary_PID = video1_pid,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 3, # mpeg2 audio stream type
            elementary_PID = audio1_pid,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 5, # AIT stream type
            elementary_PID = ait1_pid,
            element_info_descriptor_loop = [
                application_signalling_descriptor(
                    application_type = 0x0010, # HbbTV service
                    AIT_version = 1, # current ait version
                ),
            ],
        ),
    ],
    version_number = 1, # you need to change the table number
    every time you edit, so the decoder will compare its version with the
    new one and update the table
    section_number = 0,
    last_section_number = 0,
)

#
# Service Description Table (ETSI EN 300 468 5.2.3)
# this is a basic SDT with the minimum descriptors, OpenCaster has a
# big library ready to use
#

sdt = service_description_section(
    transport_stream_id = demo_transport_stream_id,
    original_network_id = demo_original_network_id,
    service_loop = [
        service_loop_item(
            service_ID = demo_service_id,
            EIT_schedule_flag = 0, # 0 no current even
information is broadcasted, 1 broadcasted
            EIT_present_following_flag = 0, # 0 no next
event information is broadcasted, 1 is broadcasted
            running_status = 4, # 4 service is running, 1
not running, 2 starts in a few seconds, 3 pausing
            free_CA_mode = 0, # 0 means service is not
scrambled, 1 means at least a stream is scrambled
            service_descriptor_loop = [
                service_descriptor(

```

```

        service_type = 1, # digital
television service
        service_provider_name =
"mediatvcom",
        service_name = "openHbb",
    ),
],
),
],
    version_number = 1, # you need to change the table number
every time you edit, so the decoder will compare its version with the
new one and update the table
    section_number = 0,
    last_section_number = 0,
)
#
# Application Informaton Table (ETSI TS 101 812 10.4.6)
#
#
ait = application_information_section(
    application_type = 0x0010,
    common_descriptor_loop = [
        external_application_authorisation_descriptor(
            application_identifiers =
[[organisationId_1,applicationId_1] ,
[organisationId_2,applicationId_2]],
            application_priority = [
                5
                ,
                1
            ]
            # This descriptor informs that 2 applications are
available on the program by specifying the applications identifiers
(couple of organization_Id and application_Id parameters) and their
related priorities (5 for the first and 1 for the second).
            # Actually our service contains only one application
so this descriptor is not relevent and is just here to show you how to
use this descriptor.
            # This descriptor is not mandatory and you could
remove it (i.e. common_descriptor_loop = []).
        )
    ],
    application_loop = [
        application_loop_item(
            organisation_id = organisationId_1, # this is a demo
value, dvb.org should assign an unique value
            application_id = applicationId_1,

            application_control_code = 1,
                # 2 is PRESENT, the decoder will
add this application to the user choice of application
                # 1 is AUTOSTART, the application
will start immediatly to load and to execute
                # 7 is DISABLED, The application
shall not be started and attempts to start it shall fail.
                # 4 is KILL, it will stop execute
the application
            application_descriptors_loop = [
                transport_protocol_descriptor(
                    protocol_id = 0x0003, # HTTP transport
protocol
                    URL_base = appli_root,
                    URL_extensions = [],

```

```

        transport_protocol_label = 3, # HTTP
transport protocol
    ),
    application_descriptor(
        application_profile = 0x0000,
            #0x0000 basic profile
            #0x0001 download feature
            #0x0002 PVR feature
            #0x0004 RTSP feature
        version_major = 1, # corresponding to
version 1.1.1
            version_minor = 1,
            version_micro = 1,
            service_bound_flag = 1, # 1 means the
application is expected to die on service change, 0 will wait after
the service change to receive all the AITs and check if the same app
is signalled or not
            visibility = 3, # 3 the applications is
visible to the user, 1 the application is visible only to other
applications
            application_priority = 1, # 1 is lowset,
it is used when more than 1 applications is executing
            transport_protocol_labels = [3], # If
more than one protocol is signalled then each protocol is an
alternative delivery mechanism. The ordering indicates

            # the broadcaster's view of which transport connection will
provide the best user experience (first is best)
        ),
        application_name_descriptor(
            application_name = appli_name,
            ISO_639_language_code = "FRA"
        ),

        simple_application_location_descriptor(initial_path_bytes =
appli_path),
    ]
),

],
    version_number = 1,
    section_number = 0,
    last_section_number = 0,
)

#
# PSI marshalling and encapsulation
#
out = open("./nit.sec", "wb")
out.write(nit.pack())
out.close
out = open("./nit.sec", "wb") # python flush bug
out.close
os.system('/usr/local/bin/sec2ts 16 < ./nit.sec > ./nit.ts')

out = open("./sdt.sec", "wb")
out.write(sdt.pack())
out.close
out = open("./sdt.sec", "wb") # python flush bug
out.close

```

```
os.system('/usr/local/bin/sec2ts 17 < ./sdt.sec > ./sdt.ts')

out = open("./pat.sec", "wb")
out.write(pat.pack())
out.close
out = open("./pat.sec", "wb") # python flush bug
out.close
os.system('/usr/local/bin/sec2ts 0 < ./pat.sec > ./pat.ts')

out = open("./pmt.sec", "wb")
out.write(pmt.pack())
out.close
out = open("./pmt.sec", "wb") # python flush bug
out.close
os.system('/usr/local/bin/sec2ts ' + str(pmt1_pid) + ' < ./pmt.sec >
./pmt.ts')

out = open("./ait.sec", "wb")
out.write(ait.pack())
out.close
out = open("./ait.sec", "wb") # python flush bug
out.close
os.system('/usr/local/bin/sec2ts ' + str(ait1_pid) + ' < ./ait.sec >
./ait.ts')

os.system('rm *.sec') # deleting of the section files.
```

Annex F. Funcions per a gestionar cookies

```
function createCookie(name, value, mins){
    if (mins) {
        var date = new Date();
        date.setTime(date.getTime()+(mins*60*1000));
        var expires = "; expires="+date.toUTCString();
    }
    else var expires = "";
    document.cookie = name+"="+value+expires+"; path=/";
}
```

```
function readCookie(name){
    var ca = document.cookie.split(';');
    var nameEQ = name + "=";
    for(var i=0; i < ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1,
c.length); //delete spaces
        if (c.indexOf(nameEQ) == 0) return
c.substring(nameEQ.length, c.length);
    }
    return null;
}
```

```
function eraseCookie(name){
    createCookie(name, "", -1);}
}
```

Annex G. Codi de configuració del widget de Twitter

```
<div id="twitter" style="position: absolute; z-index: 0; top:
330px; left: 40px;">
  <script charset="utf-8" src="Twitter.js"></script>
  <script>
    new TWTR.Widget({
      version: 2,
      type: 'search',
      search: '#hbbtv',
      interval: 5000,
      title: 'Participa!',
      subject: '#hbbtv',
      width: 350,
      height: 250,
      theme: {
        shell: {
          background: '#8e1da',
          color: '#ffffff'
        },
        tweets: {
          background: '#ffffff',
          color: '#444444',
          links: '#1985b5'
        }
      },
      features: {
        scrollbar: false,
        loop: true,
        live: true,
        behavior: 'default'
      }
    }).render().start();
  </script>
</div>
```


Annex H. Captures de les diferents versions de l'aplicació HbbTV desenvolupada

Versió completa amb FireHbbTV



Fig. H.1.1 ACCIÓ: Entrem a l'aplicació. Focus a "Alineació FCB"



Fig. H.1.2 ACCIÓ: Premem VK_DOWN. Focus a "Alineació RMCF"



Fig. H.1.3 ACCIÓ: Premem VK_DOWN. Focus a “Estadístiques”



Fig. H.1.4 ACCIÓ: Premem VK_DOWN. Focus a “Minut a minut”

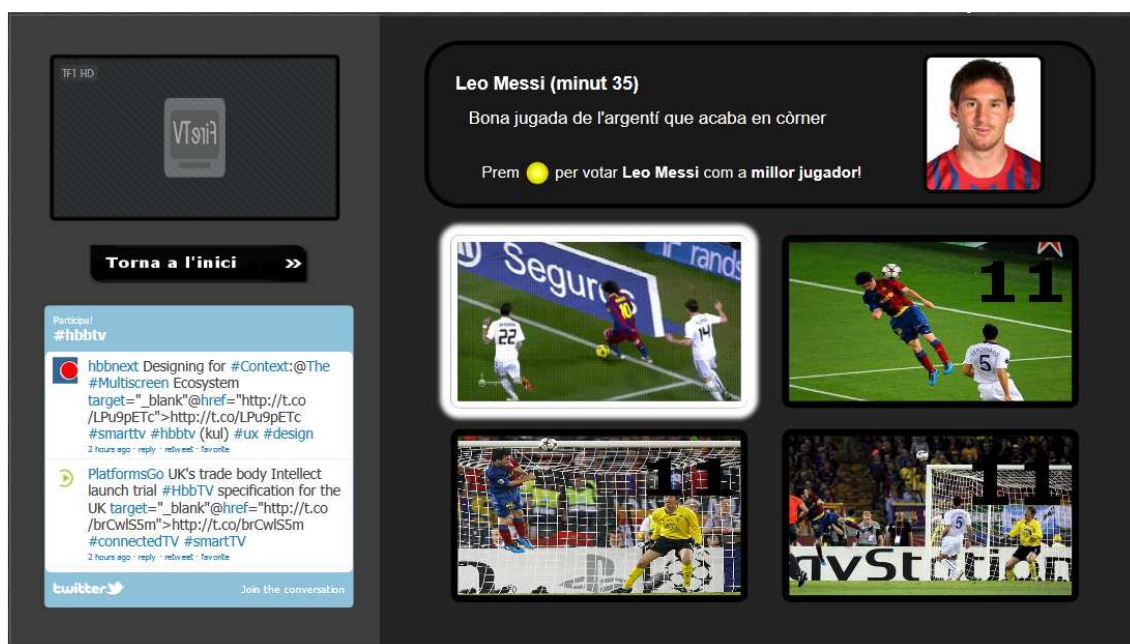


Fig. H.1.5 ACCIÓ: Seleccionem una jugada⁸

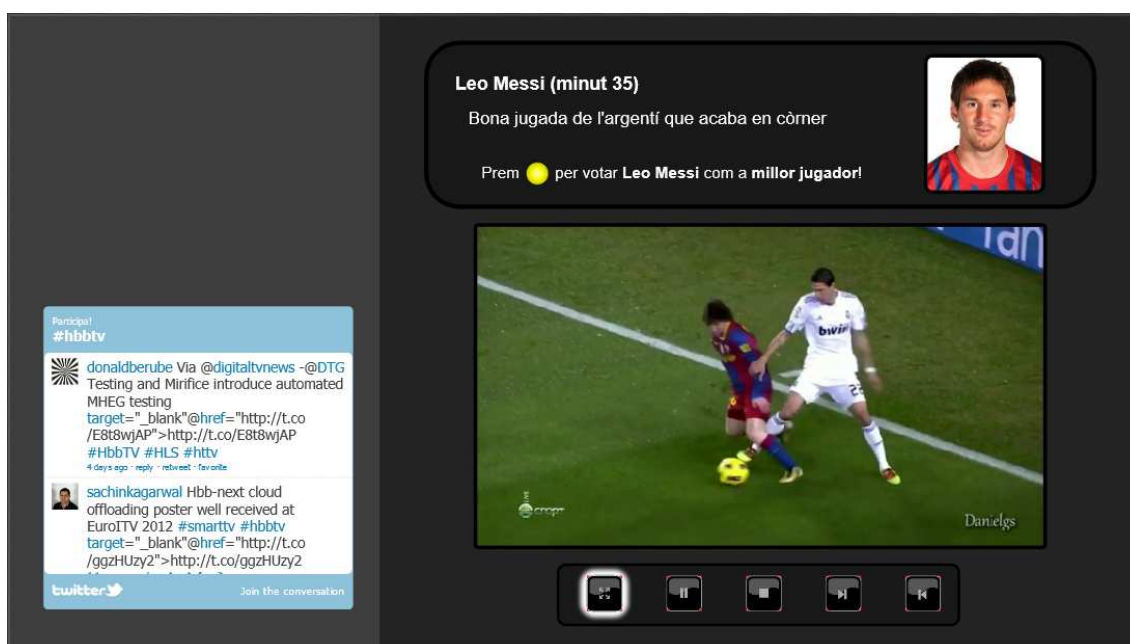


Fig. H.1.6 ACCIÓ: Seleccionem un dels quatre vídeos de la jugada

⁸ GIF animat a la jugada que té el focus

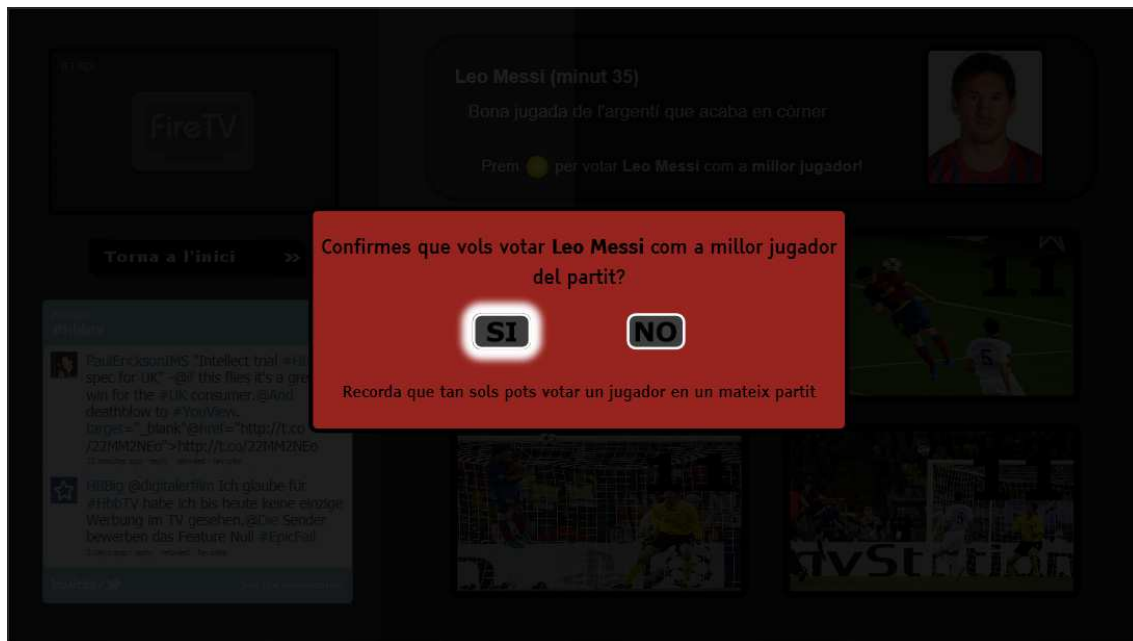


Fig. H.1.7 ACCIÓ: Premem STOP (tornem a la pantalla inicial de la jugada) i premem VK_YELLOW

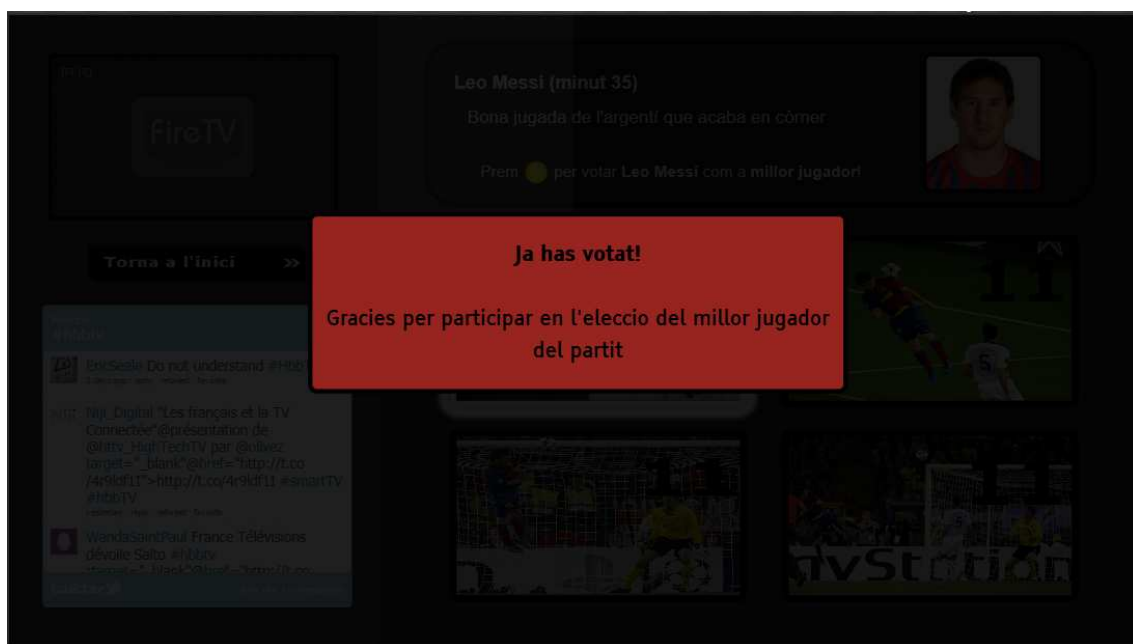


Fig. H.1.8 ACCIÓ: Premem "SI" i tornem a prémer VK_YELLOW (votar per segon cop)

Comprovem que el navegador ha guardat correctament la *cookie* del vot.

Versió completa amb Opera TV Emulator



Fig. H.2.1 ACCIÓ: Entrem a l'aplicació. Focus a "Alineació FCB"

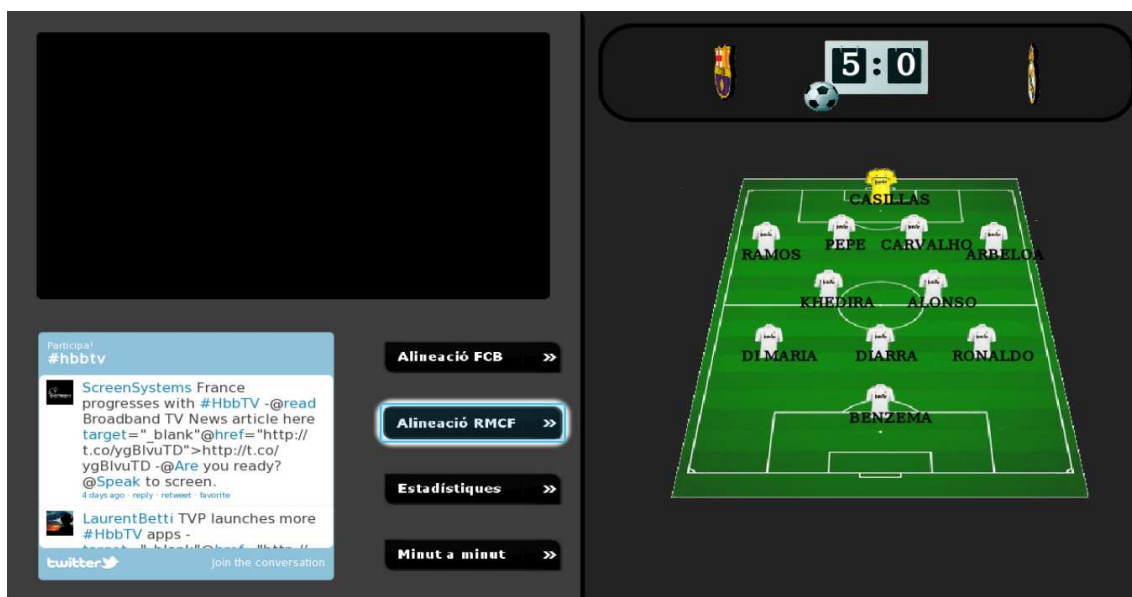


Fig. H.2.2 ACCIÓ: Premem VK_DOWN. Focus a "Alineació RMCF"



Fig. H.2.3 ACCIÓ: Premem VK_DOWN. Focus a “Estadístiques”

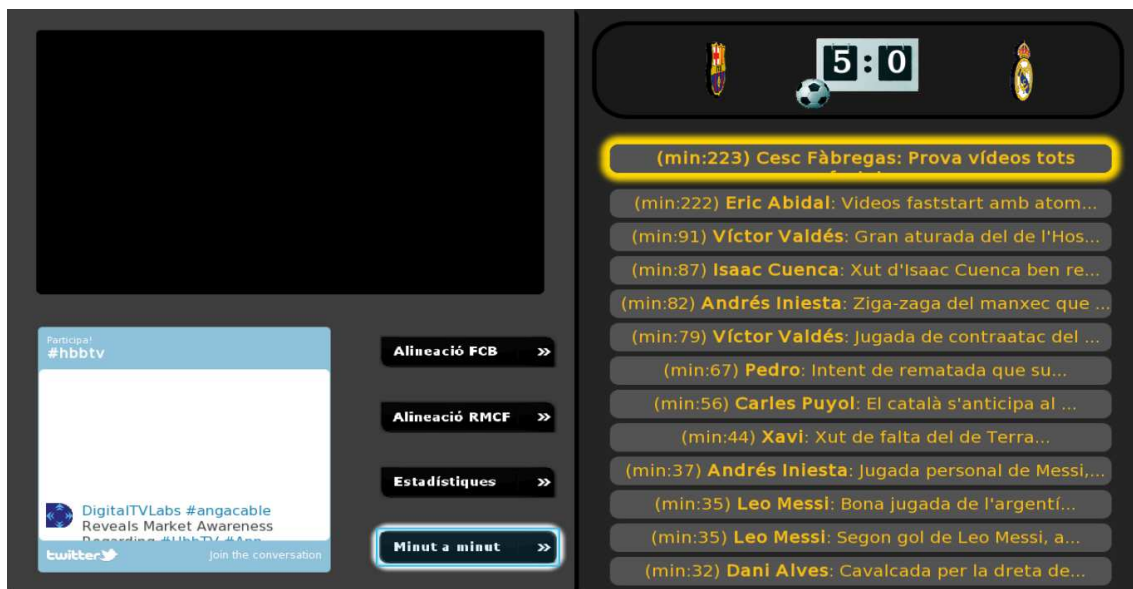


Fig. H.2.4 ACCIÓ: Premem VK_DOWN. Focus a “Minut a minut”

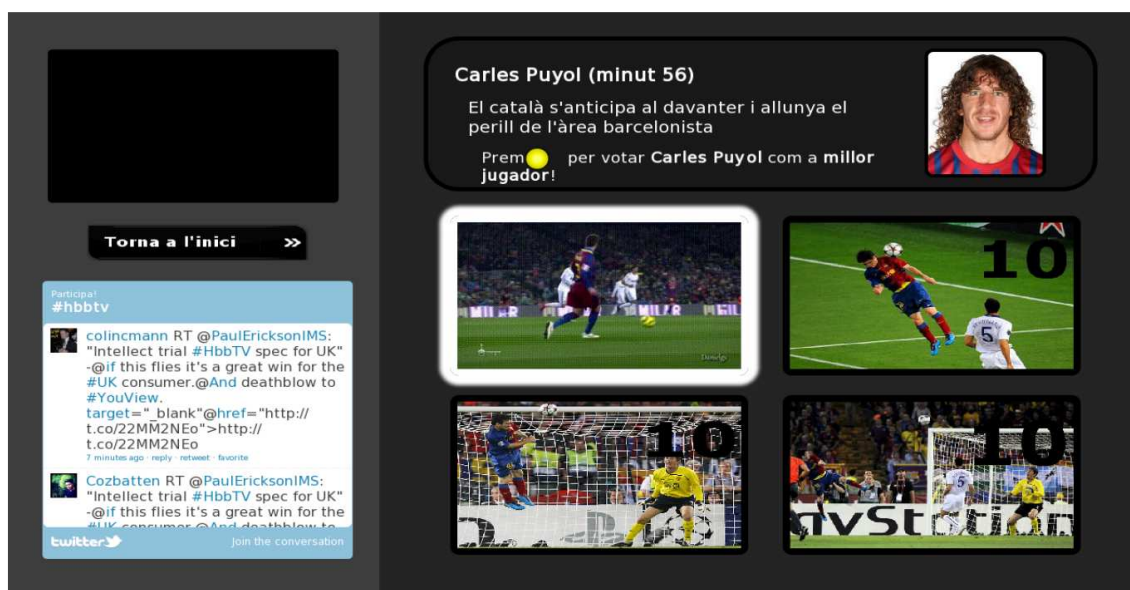


Fig. H.2.5 ACCIÓ: Seleccionem una jugada⁹



Fig. H.2.6 ACCIÓ: Seleccionem un dels quatre vídeos de la jugada

⁹ GIF animat a la jugada que té el focus



Fig. H.2.7 ACCIÓ: Premem STOP (tornem a la pantalla inicial de la jugada) i premem VK_YELLOW



Fig. H.2.8 ACCIÓ: Premem "SI" i tornem a prémer VK_YELLOW (votar per segon cop)

Comprovem que el navegador ha guardat correctament la *cookie* del vot.

Versió Engel amb EN2000



Fig. H.3.1 ACCIÓ: Entrem a l'aplicació. Focus a "Alineació FCB"

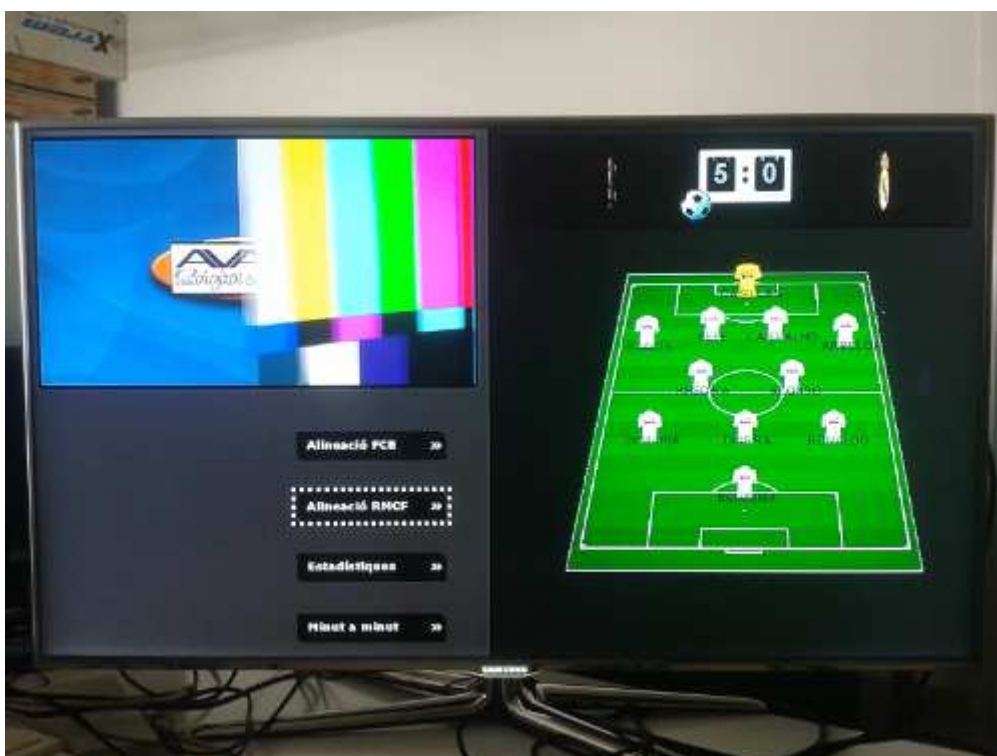


Fig. H.3.2 ACCIÓ: Premem VK_DOWN. Focus a "Alineació RMCF"



Fig. H.3.3 ACCIÓ: Premem VK_DOWN. Focus a “Estadístiques”



Fig. H.3.4 ACCIÓ: Premem VK_DOWN. Focus a “Minut a minut”

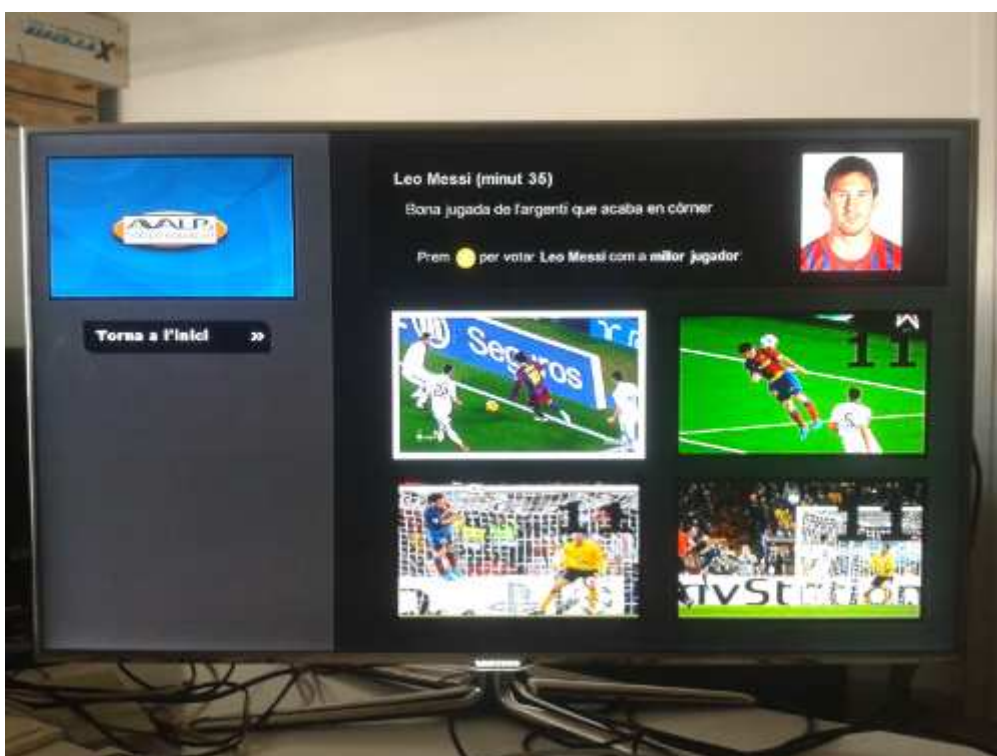


Fig. H.3.5 ACCIÓ: Seleccionem una jugada¹⁰



Fig. H.3.6 ACCIÓ: Seleccionem un dels quatre vídeos de la jugada

¹⁰ GIF animat a la jugada que té el focus



Fig. H.3.7 ACCIÓ: Premem STOP (tornem a la pantalla inicial de la jugada) i premem VK_YELLOW

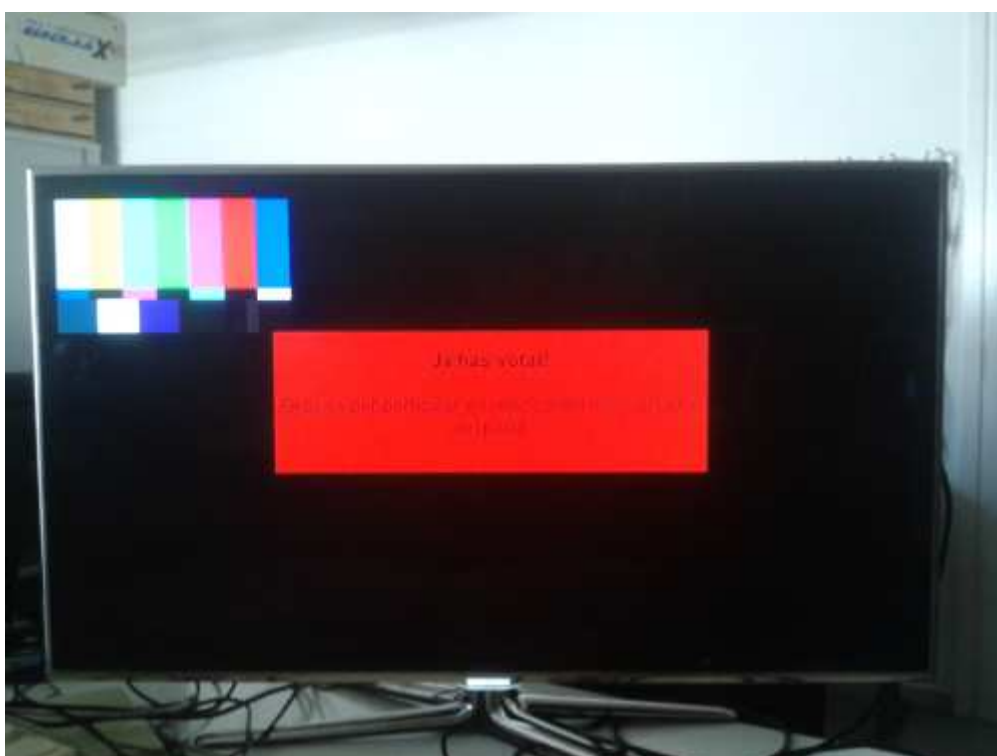


Fig. H.3.8 ACCIÓ: Premem "SI" i tornem a prémer VK_YELLOW (votar per segon cop)

Comprovem que el navegador ha guardat correctament la *cookie* del vot.

Versió Samsung amb Samsung SmartTV



Fig. H.4.1 ACCIÓ: Entrem a l'aplicació. Focus a "Alineació FCB"

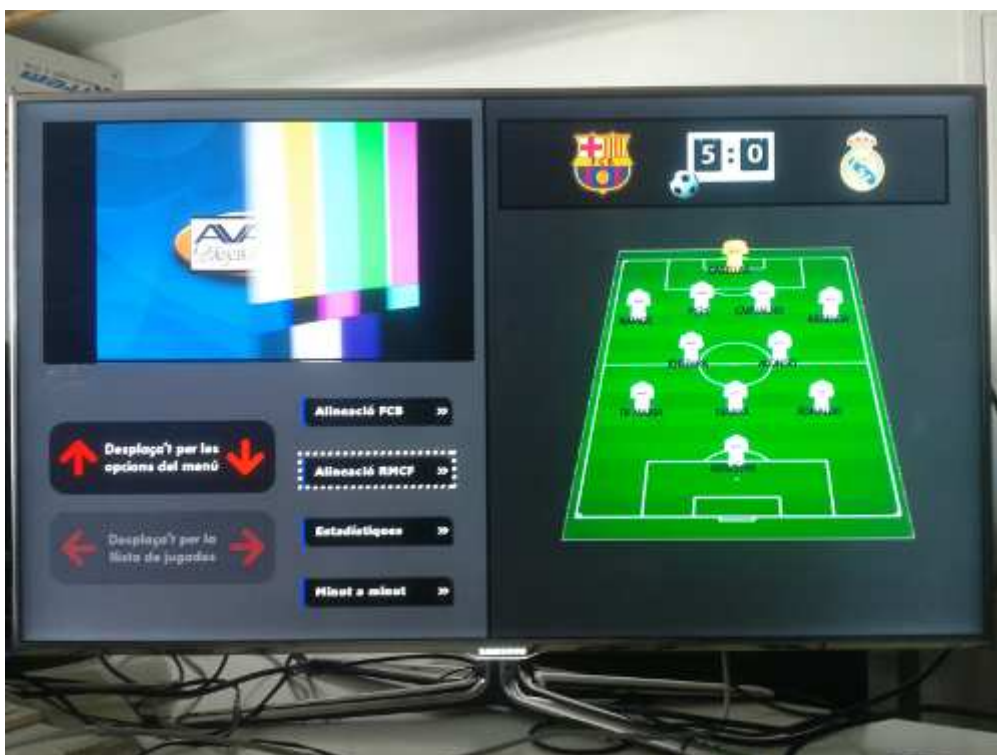


Fig. H.4.2 ACCIÓ: Premem VK_DOWN. Focus a "Alineació RMCF"



Fig. H.4.3 ACCIÓ: Premem VK_DOWN. Focus a “Estadístiques”



Fig. H.4.4 ACCIÓ: Premem VK_DOWN. Focus a “Minut a minut”



Fig. H.4.5 ACCIÓ: Seleccionem una jugada¹¹

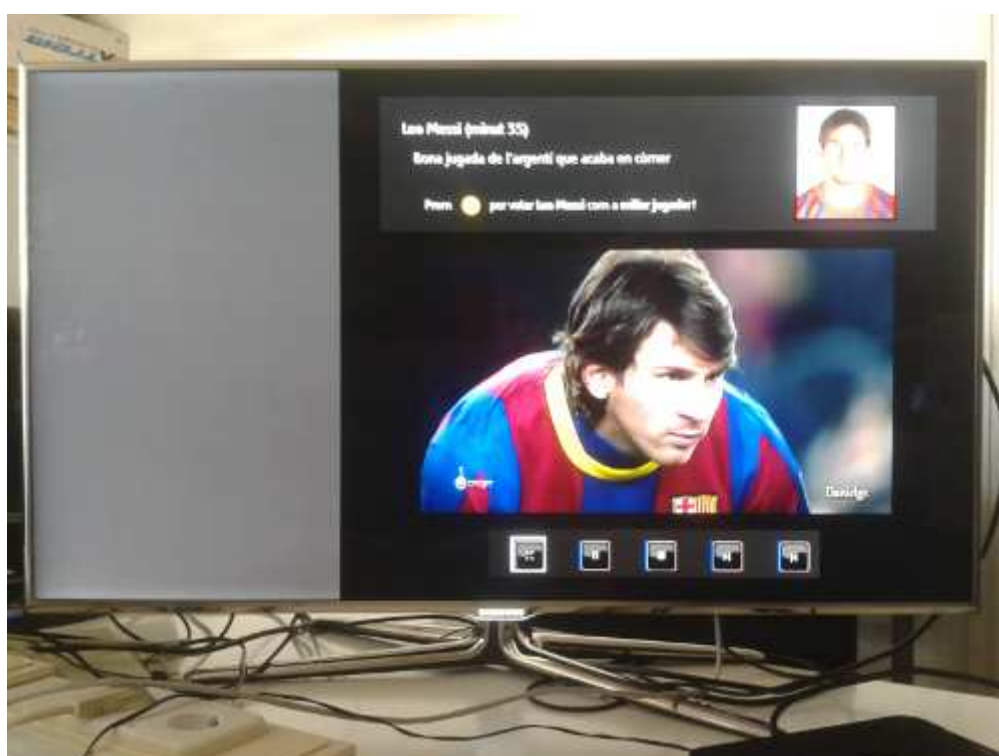


Fig. H.4.6 ACCIÓ: Seleccionem un dels quatre vídeos de la jugada

¹¹ No es mostra cap GIF animat a la jugada seleccionada

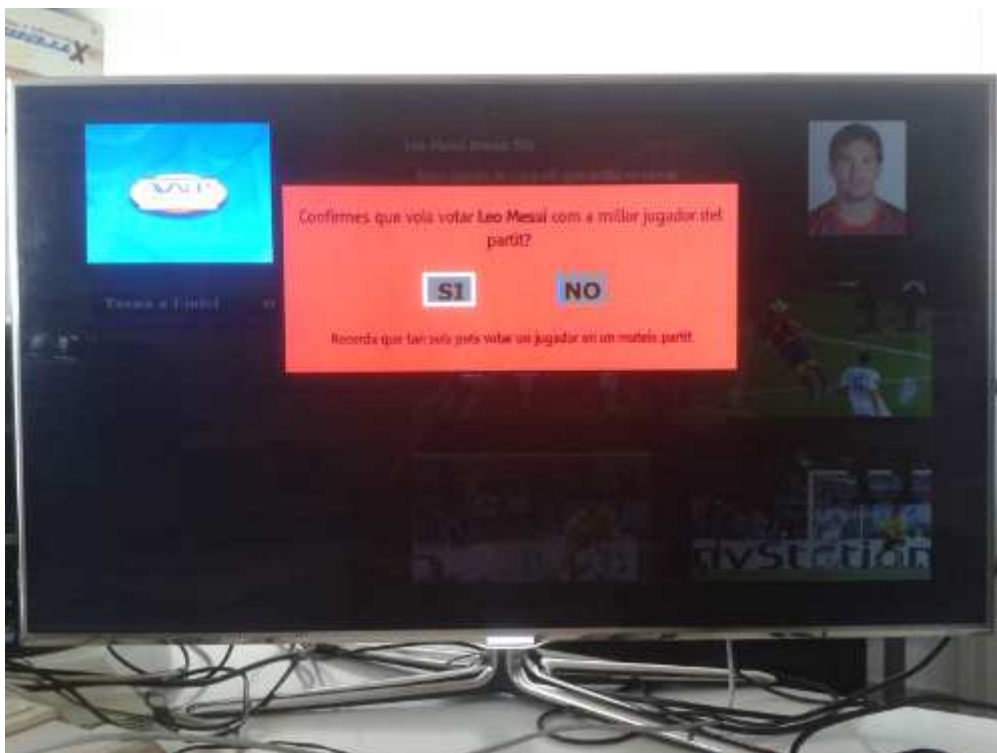


Fig. H.4.7 ACCIÓ: Premem STOP (tornem a la pantalla inicial de la jugada) i premem VK_YELLOW

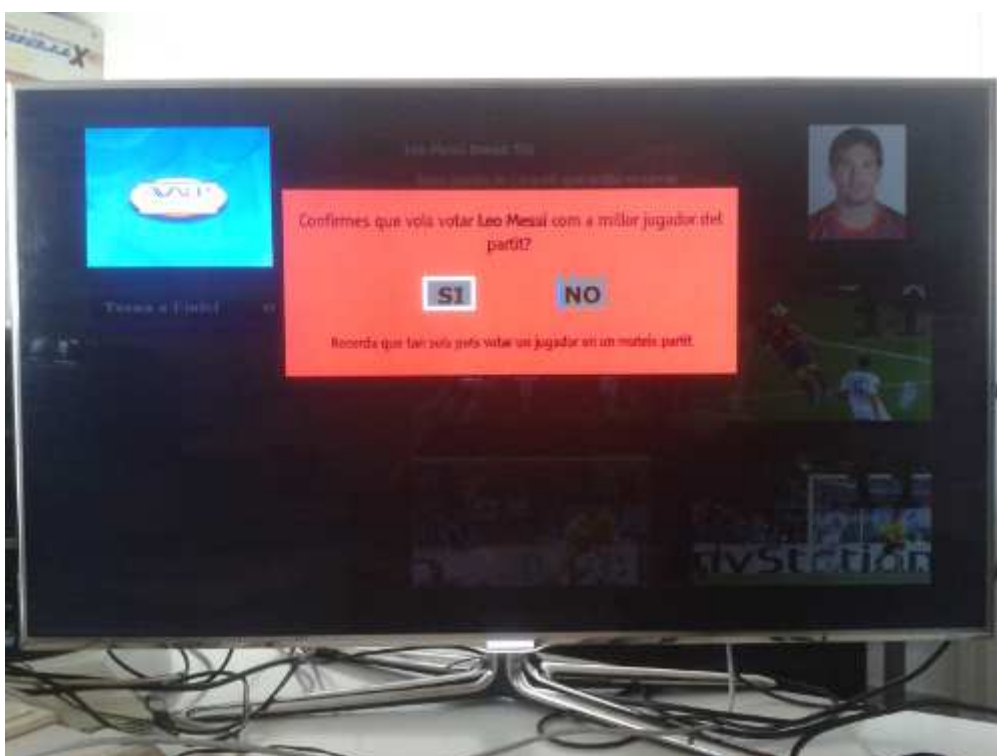


Fig. H.4.8 ACCIÓ: Premem "SI" i tornem a prémer VK_YELLOW (votar per segon cop)

Comprovem que en aquest cas el navegador no ha guardat la cookie del vot.