

APLICACIÓ RICA D'INTERNET PER A LA CONSULTA AMB TEXT I IMATGE AL REPOSITORI DE VÍDEOS DE LA CORPORACIÓ CATALANA DE MITJANS AUDIOVISUALS

RAMON SALLA ROVIRA

TUTORITZAT PER XAVIER GIRÓ I XAVIER VIVES

1 MOTIVACIÓ

Fa més de dos anys vaig entrar a treballar a la Corporació Catalana de Mitjans Audiovisuais per donar suport a diferents projectes subvencionats a nivell estatal i europeu. El present document és la memòria del projecte final de carrera realitzat en aquesta empresa durant l'any 2009.

La Corporació és una empresa catalana puntera en innovació en el camp dels mitjans de comunicació destinada al servei públic per transmetre informació i entreteniment amb la màxima qualitat possible.

Entrar-hi a treballar ha suposat un revulsiu en la meva orientació com a enginyer i m'ha introduït a un sector, destinat a la producció, postproducció, emmagatzematge, cerca, indexació i distribució. Aquesta experiència ha estat, com a mínim, apassionant.

El projecte i3media¹, subvencionat per l'estat espanyol, pretén unir en un sol marc col·laboratiu empreses punteres del sector per tal d'impulsar els continguts intel·ligents, la personalització i els processos automàtics.

Comptar dins d'aquest projecte amb el grup de recerca de processat de la imatge de la Universitat Politècnica de Barcelona (GPI) és una idoneïtat que em permet mantenir les arrels amb la meva universitat. Aquest fet, em porta a tirar endavant aquest projecte final de carrera que culmina la meva aspiració de ser enginyer i permet aportar el meu humil gra d'arena a l'ambiciós projecte del que tinc l'orgull i la sort de formar part.

¹ www.i3media.org

2 ÍNDEX

4	Introducció.....	7
5	Estat de l'Art	9
5.1	Consulta i recuperació de fitxers digitals.....	9
5.1.1	Problema general, buscant solució a la bretxa semàntica	9
5.1.2	Recuperació de vídeos.....	11
5.1.3	Descriptors d'un objecte multimèdia	12
5.1.4	Tipus de consulta	13
5.2	Interfícies Web Enriquides.....	23
5.2.1	Aplicacions riques d'internet i serveis web	23
5.2.2	Càlcul als Núvols a l'empresa	24
	SaaS (software as a service)	25
	Utility computing	25
	Web services in the cloud.....	25
5.2.3	Definició del concepte d'aplicació web enriquida.....	27
5.2.4	Diagrames comparatius.....	28
5.2.5	Avantatges i inconvenients de les RIAs	29
5.2.6	Entorns de programació de RIAs (<i>FrameWorks</i>)	30
5.2.7	Persistència de les RIAs	31
5.2.8	Servidors	33
5.2.9	Protocols de comunicació	34
6	Escenaris d'ús i requeriments d'usuari inicials.....	36
6.1	Escenaris d'ús.....	36
6.2	Requisits del Sistema	38
7	Arquitectura del Sistema	40
7.1	Introducció	40
7.2	Entorn de programació emprat	41
7.3	Descripció dels recursos en el pla servidor.....	42
7.3.1	Gràfic General.....	43
7.3.2	A3k6 (1), el servidor nucli	44

7.3.3	Apache Solr (4), recuperació optimitzada de metadades	44
7.3.4	Repositori d'Images (3), l'extracció d'imatges mitjançant l'FFMPEG	46
7.3.5	Servei Web de Yahoo! (3)	48
7.3.6	Servei Web de l'Upseek	49
7.3.7	Problemes de seguretat del connector Flash per a serveis externs	50
7.3.8	Connexió amb el client, implementació del protocol AMF	52
7.4	Descripció dels recursos en el pla client	54
7.4.1	Introducció al Digimatge	54
7.4.2	Model del programa Client	54
7.4.3	Beans.xml, configuració de les comunicacions i els controladors	56
7.4.4	appModel, objectes persistents durant la sessió de l'aplicació	57
7.4.5	Interfície de l'aplicació	58
8	Test d'Usuari	61
8.1	Introducció	61
8.2	Impressions dels usuaris	61
9	Conclusions i Treball Futur	63
9.1	Conclusions	63
9.1.1	Un projecte d'empresa	63
9.1.2	Objectius complerts	63
9.1.3	Altres resultats aconseguits	65
9.2	Treball futur	66
9.2.1	Servidor	66
9.2.2	Eina Apache Solr	67
9.2.3	Eina Client	67
9.2.4	Interfície per a la detecció d'entitats	68
10	Annexes	69
11	Bibliografia	70

3 FIGURES

Figura 1 Requeriments més valorats	8
Figura 2 Exemple de consulta mitjançant descriptors	10
Figura 3 Microsoft BING	14
Figura 4 Google Images	14
Figura 5 Mnemomap	15
Figura 6 Doc. Img. Retrieval System	15
Figura 7 Multicolor	15
Figura 8 G.O.S.	16
Figura 9 ImgSeek.....	16
Figura 10 Simplicity.....	16
Figura 11 cuZero	17
Figura 12 Behold.....	18
Figura 13 Google Similar Images	18
Figura 14 Google Swirl.....	18
Figura 15 Picitup	19
Figura 16 Mirror.....	19
Figura 17 Sapir	19
Figura 18 Xcavator	20
Figura 19 Scketch2Photo	20
Figura 21 Pixolu	20
Figura 22 Google Picasa.....	21
Figura 23 G.A.T.	21
Figura 24 Rellevància del Cloud Computing	24
Figura 25 Variants dels Serveis Web	25
Figura 26 onLive Service	26
Figura 27 Procés de Web Tradicional	28
Figura 28 Procés d'Aplicació Web Enriquida	28
Figura 29 Adobe Kuler en les versions web, escriptori i giny	32
Figura 30 Ús dels principals servidors (Font: Netcraft)	33
Figura 31 Digiton.....	37
Figura 32 Descripció d'un asset i un estrat.....	40
Figura 33 Diagrama de Recursos del Sistema.....	43
Figura 34 Configuració de l'Apache Solr	45
Figura 35 Exemple de crida REST a Solr.....	46
Figura 36 Extractor de Fotogrames	47
Figura 37 Comanda FFMPEG utilitzada	48

Figura 39 Exemple d'alternatives	49
Figura 38 XML de resposta al servei de Yahoo!	49
Figura 40 Resposta al mètode QueryByExample	50
Figura 41 Accedint a l'aplicació, creant un camí segur.....	51
Figura 42 Recuperació correcta d'informació externa	51
Figura 43 Arquitectura de les comunicacions amb BlazeDS	52
Figura 44 Proxy-config.xml	53
Figura 45 Remoting-config.xml.....	53
Figura 46 Digimatge.....	54
Figura 48 Relació entre els mòduls del programa.....	55
Figura 47 Estructura de fitxers del programa.....	55
Figura 49 Recomanació de paraules.....	57
Figura 50 Digimatge, pantalla principal.....	58
Figura 51 Detall de l'asset.....	59
Figura 52 Detecció d'Artur Mas.....	60

4 INTRODUCCIÓ

La cerca de continguts en empreses relacionades amb la producció audiovisual s'ha convertit en un problema creixent degut a la gran quantitat de dades que s'hi generen gràcies al creixent ample de banda, els codificadors de vídeo més eficients i la potència dels ordinadors de consum. El procés de digitalització dels senyals audiovisuals viscut en la darrera dècada ha incrementat el ritme de creació de continguts i en fa cada cop més inviable la seva indexació manual.

Les eines actuals de processat d'imatge estan avançant per solucionar l'anomenada bretxa semàntica (*semantic gap*) entre els bits dels continguts i els conceptes semàntics que representen. Aquesta bretxa es dona quan s'intenten computeritzar les definicions d'objectes i idees per a solucionar consultes aparentment tan senzilles com "troba imatges de la Casa Blanca a on hi aparegui el president dels Estats Units". L'objectiu és facilitar l'accés dels usuaris a la base de dades mitjançant indexacions automàtiques i nous mètodes per formular consultes i executar cerques.

Una de les línies d'investigació en aquest camp proposa evolucionar la clàssica consulta textual (avui en dia encara imprescindible) cap a una nova manera de recuperar continguts. L'objectiu és desenvolupar una metodologia més rica, per exemple, representant idees amb imatges i llençar noves consultes a partir d'aquestes imatges, les seves regions o un esbós del concepte que té a la ment l'usuari que busca continguts.

Aquesta nova modalitat de consulta requereix d'eines de processat que n'extreguin noves característiques més enllà de les clàssiques paraules clau. També es requereixen noves interfícies gràfiques d'usuari que despleguin el potencial dels nous descriptors basats en les característiques perceptuals de les imatges (el color, la textura, la forma...), més en consonància amb el funcionament de la ment humana. Les noves interfícies han de respondre a la flexibilitat i a l'agilitat amb què la ment humana crea, processa i descarta els resultats mostrats pels motors de cerca.

El present projecte persegueix crear una interfície gràfica d'usuari que sense pretendre trencar amb les consultes per text, busca millorar-les amb l'ús de nous descriptors avançats de l'estat de l'art en processat d'imatge. Per fer-ho, l'eina proposa una nou mètode de consulta i filtratge de vídeos a partir dels seus fotogrames i de les metadades textuales que tenen annexades. D'aquesta forma es pretén donar resposta a alguns dels problemes amb què se solen trobar professionals del món audiovisual, com poden ser periodistes o documentalistes. La Figura 1 mostra els resultats d'una

enquesta a usuaris professionals de la Corporació Catalana de Mitjans Audiovisuals (CCMA) en el marc del projecte I3Media². En la gràfica s'aprecia que dins d'un ventall de requeriments, els usuaris valoren una ordenació intel·ligent dels resultats d'una eina de consulta.

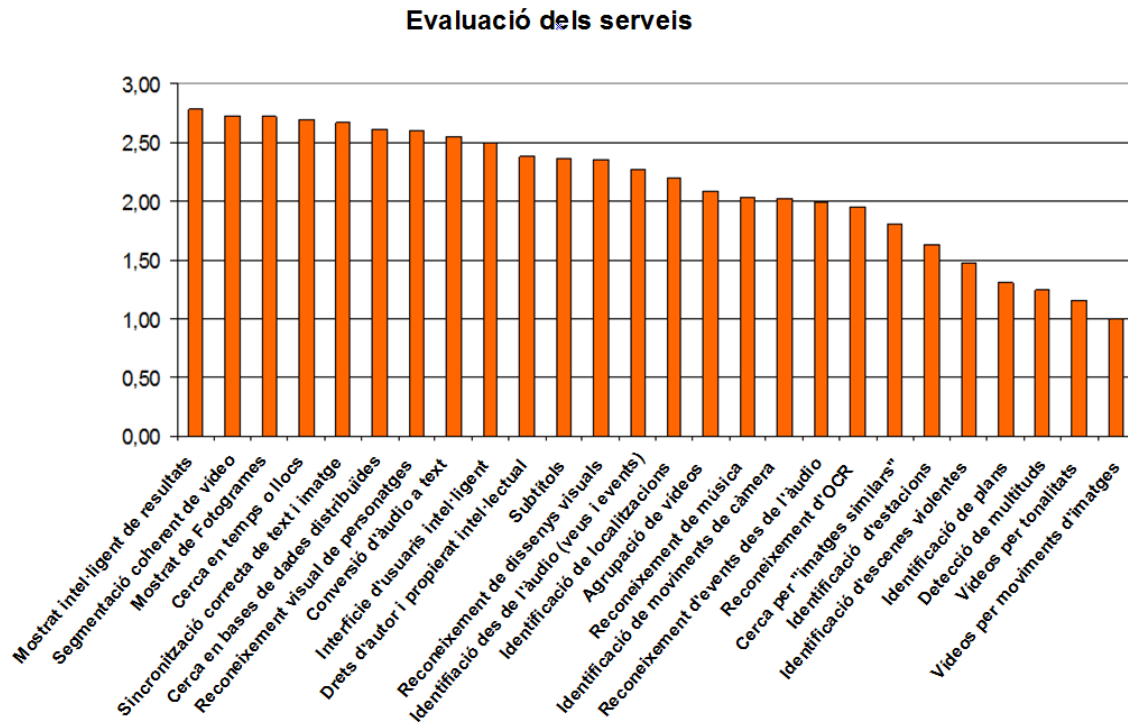


FIGURA 1 REQUERIMENTS MÉS VALORATS

² www.i3media.org

5 ESTAT DE L'ART

5.1 CONSULTA I RECUPERACIÓ DE FITXERS DIGITALS

5.1.1 PROBLEMA GENERAL, BUSCANT SOLUCIÓ A LA BRETXA SEMÀNTICA

La **cerca** de fitxers digitals com vídeos, imatges, arxius de so, documents de text... és una tasca complexa que inclou un procés d'extracció de paràmetres rellevants dels objectes a **cercar**, una **consulta** mitjançant eines que treballin amb aquests descriptors i finalment una **navegació** sobre els resultats obtinguts.

L'origen d'una consulta es troba en la ment de l'usuari. L'usuari pensa en un concepte, és a dir, una idea formada a la ment que necessita avaluar en una base de dades. No sempre aquest concepte està definit i clar. Pot ser que l'usuari sàpiga amb molta precisió el que està cercant o potser que aquesta idea es vagi fent més nítida a mesura que l'eina de consulta li va presentant resultats recomanats intel·ligentment.

En aquest punt es planteja un dels principals problemes que es troben els usuaris de les eines de consulta: la **bretxa semàntica**. A [Smeulders et al.]ⁱ es defineix com la falta de coincidència entre, per una banda, la interpretació que un usuari fa d'un concepte en un context determinat, i per altra banda, la informació que es pot extreure dels **continguts digitals** que representen aquest concepte.

L'eina de cerca no deixa de ser un programa informàtic que no entén de conceptes mentals. Així doncs, cal un pas entremig d'extracció de **descriptors** que puguin ser processats per una màquina per avaluar-ne la semblança. Aquests descriptors poden ser de diferent naturalesa, com per exemple una paraula clau, una imatge o regió de la mateixa, una textura, un color, un esbós, etc.

Un cop escollit tant el concepte mental, com els descriptors que millor descriuen aquest concepte, es llença una consulta sobre una base de dades de descriptors que prèviament s'han extret dels objectes digitals. Lògicament, per tal que la consulta sigui viable, els descriptors de la consulta han de ser de la mateixa naturalesa que els descriptors de la base de dades. La Figura 2 mostra un exemple de consulta mitjançant dos descriptors diferents, un textual i un visual.

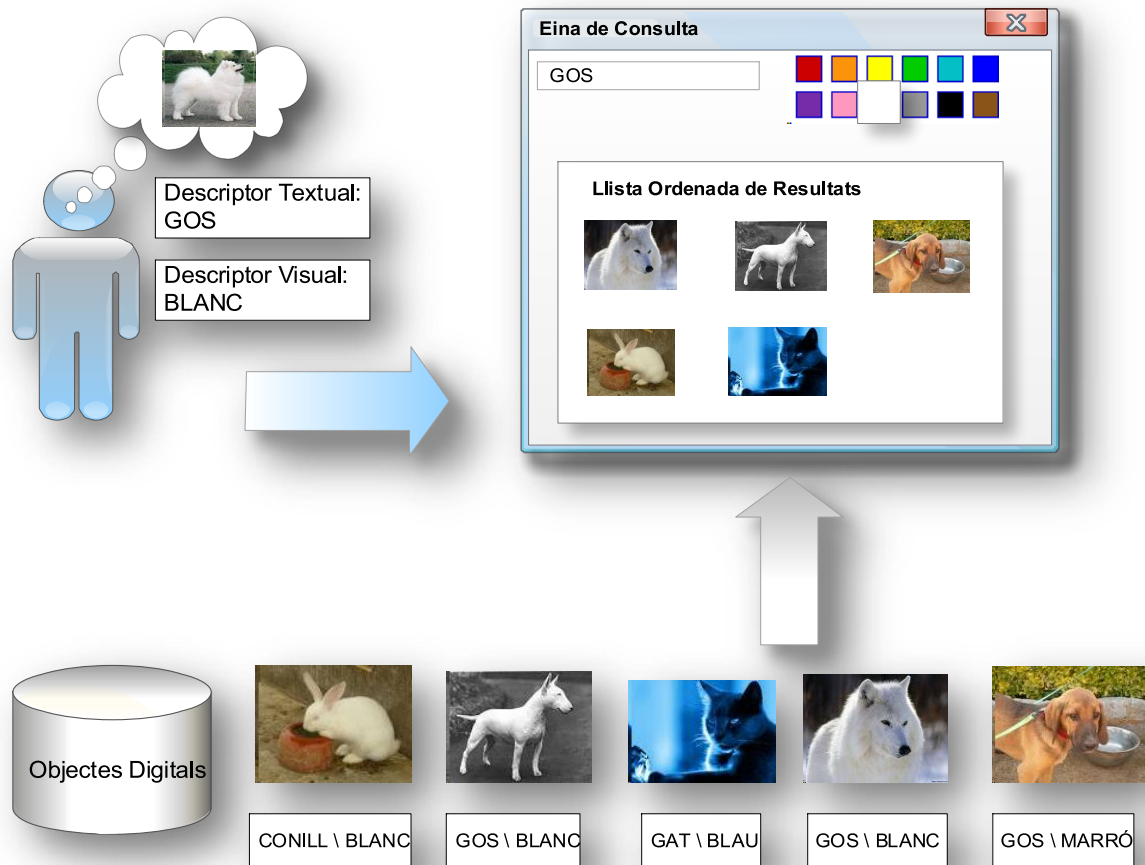


FIGURA 2 EXEMPLE DE CONSULTA MITJANÇANT DESCRIPTORS

La **consulta** també contempla un procés manual de navegació per aquests resultats obtinguts, mostrant percentatges d'encert, rellevància de resultats, reproductors dels fitxers media, taules amb les metadades annexades.... A [Hare et al.]ⁱⁱ es comenten diferents intents en la cerca de continguts que han fallat en la forma de presentar els resultats tenint en compte la subjectivitat de l'usuari final, que és qui acaba decidint si els resultats trobats s'adeqüen o no al concepte que s'estava buscant. De fet, les interfícies ben dissenyades dels programes de consulta són un dels punts més útils per a solucionar el problema que suposa la **bretxa semàntica**.

La **bretxa semàntica** és doncs una de les dificultats per a saber indexar automàticament sobre bases de dades, però a [Inoue]ⁱⁱⁱ també trobem altres impediments que fan de la cerca i la consulta d'objectes un repte digne del segle en què ens trobem més enllà de la extracció de característiques visuals. Alguns d'aquests impediments són la dificultat d'extreure característiques bàsiques pels usuaris com són el temps, l'espai, la importància dels esdeveniments, l'abstracció i la emoció, o imitar la facilitat amb la que les persones suprimeixen característiques irrellevants. Altres impediments són la llengua amb què es decideix anotar el contingut en cas de voler-lo recuperar mitjançant paraules clau textuales. També, i no menys important, la

motivació dels usuaris és un factor clau per simplificar la seva tasca de consulta. Aquesta tasca pot ser o bé ocasional durant el transcurs d'un dia de treball o bé completa. La tasca ocasional és el cas d'alguns periodistes no especialitzats en documentació que només accedeixen a l'eina per fer una consulta ràpida "tots els vídeos del dia d'avui on hi aparegui la paraula *medi ambient*". La tasca completa durant el transcurs del dia és el cas de documentalistes especialitzats els quals han de donar resposta a peticions complicades a d'altres periodistes com "cerca una imatge on hi aparegui el president Obama i la seva família vestits de gala".

5.1.2 RECUPERACIÓ DE VÍDEOS

La consulta dins del món audiovisual modelitza un procés on un documentalista, reporter o usuari anàleg del món audiovisual, busca segments de fitxers contenint persones, events, localitzacions, objectes, fragments de text,... d'interès. Aquesta consulta es realitza mitjançant descriptors extrets sobre algun element d'un vídeo, com ara les seves metadades, o algun dels fotogrames del mateix. Cal tenir present que en el cas de cadenes de televisió o empreses de postproducció, la finalitat de la consulta és recuperar vídeos per a la seva edició i emissió. El pas de consulta de vídeos mitjançant els seus fotogrames és possible gràcies a eines que extreuen intel·ligentment els fotogrames més representatius dels vídeos. Normalment, aquestes eines d'extracció de fotogrames són part del anomenats MAMs (Media Asset Management), bases de dades especialitzades per arxius de vídeo en les empreses del sector audiovisual, que contemplen des de la ingesta del material a la base de dades fins la gestió de tot el seu contingut.

Tornant al problema de la **bretxa semàntica** específic a imatges, tenim tres formes diferents de fer-hi un pont: deixar als usuaris anotar les imatges, idear un sistema automàtic per a l'anotació de les imatges o bé idear un sistema que funcioni amb els descriptors que no requereixin cap interpretació semàntica. La primera opció és la que s'ha practicat fins ara, però la creixent quantitat d'imatges que es generen fa cada cop més complicat aquest etiquetatge manual. La via a desenvolupar és doncs completar les descripcions dels objectes amb descriptors visuals extrets automàticament i la presentació dels resultats mitjançant eines eficaces i intuïtives.

5.1.3 DESCRIPTORS D'UN OBJECTE MULTIMÈDIA

És important entendre quins tipus de descriptors podem aconseguir dels arxius audiovisuals com els fitxers de vídeo, els documents de text o els fitxers d'àudio que haurem de cercar. Per exemple, una cançó dura tres minuts amb trenta-dos segons, una piràmide té una forma triangular i el president Obama té un color de pell fosc.

Hi ha cert consens que qualsevol eina de cerca d'imatges està lligada intrínscament amb els descriptors de què es disposa per a la seva cerca. Així doncs, cal definir primer a què ens referim com a descriptors i quins són els descriptors més emprats en les eines de cerca actuals.

Podem separar els descriptors segons el seu nivell conceptual baix, mitjà o alt.

Els **descriptors baix nivell**, són aquells que utilitzen la informació inherent únicament i exclusiva al contingut entesa com a un conjunt binari. Alguns exemples d'aquests descriptors són la informació a nivell de píxel que ofereix la imatge, histogrames de color, luminància, saturació o tonalitat. Les tècniques de processament de la imatge com les segmentacions de les imatges i les seves múltiples transformades aporten més eines per a la extracció d'aquest tipus de descriptors.

Els **descriptors de nivell mitjà**, són aquells que utilitzen la informació de context de la imatge o que es troben durant la seva generació. Així per exemple, el nom de la imatge, la data de la seva creació o obtenció, informació EXIF (Exchangeable Image Format)³ d'una fotografia o informació ID3⁴ d'un àudio, la web a on s'ha trobat la imatge, el vídeo a la que correspon...

Finalment, els **descriptors d'alt nivell** són el que es poden trobar en les *metadades* textuales annexades a l'objecte. Són els descriptors que més valor aporten actualment a la cerca d'imatges perquè són els més propers als conceptes semàntics que inicien la cerca. Per desgràcia, també són els descriptors més difícils d'obtenir. En són exemples, les etiquetes manuals amb què s'anoten les imatges, transcripcions d'àudio dels vídeos a on pertanyen aquestes imatges, text de les web a on es troben, notícia o revista a on s'ubica la imatge, *estrats* temporals que la descriuen...

A [Salembier et al.]^{iv} s'explica que els descriptors de baix nivell prendran cada cop més rellevància a mesura que es pugui automatitzar-ne la seva extracció, també es pregunta en la conveniència de combinar tots els tipus de descriptors enlloc de potenciar només els descriptors de baix nivell. La resposta recau en la gran redundància que ofereixen els descriptors de baix nivell [Eidenberger]^v i en la emergent potencialitat de la informació textual (descriptors d'alt nivell) generada

³ Exchangeable Image File Format, www.exif.org

⁴ www.id3.org

manualment gràcies al impuls de la web 2.0. Així, per exemple, a [Klamma et al.]^{vi} es defineix una metodologia d'extracció de paraules claus generades en webs generalistes d'accés públic com Youtube⁵, Flickr⁶ o Delicious⁷ i en la seva ordenació i categorització mitjançant l'estàndard MPEG-7.

Aquest estàndard, promogut per la *Moving Pictures Experts Group*⁸, permet una estandardització dels descriptors per a continguts audiovisuals. El nou format descriu una metodologia comuna per a l'emmagatzematge de descriptors en fitxers XML. També defineix el nou llenguatge DDL (*Description Definition Language*) que permet relacionar i estructurar els descriptors dels objectes audiovisuals. L'objectiu final és poder aconseguir transaccions entre empreses, organitzacions i particulars dels seus arxius juntament amb un fitxer MPEG-7. Aquest fitxer simplificarà la seva indexació, cerca i consulta des de qualsevol aplicació que entengui l'estàndard.

5.1.4 TIPUS DE CONSULTA

Un cop definits els descriptors com a les dades sobre les quals calcular la semblança en qualsevol tipus de consulta, passem a veure distintes modelitzacions d'aplicacions per a fer consultes segons la naturalesa de la font d'entrada i sortida i els descriptors que se n'extreuen, així com exemples d'aplicacions reals. Aquest apartat es centra en categoritzar i posar exemples de consultes de documents multimèdia que es poden trobar fàcilment en un repositori audiovisual, és a dir, en un MAM (Media Asset Management).

Descripció	
Consulta	Textual
Cerca	Textuals
Resultats	Textuals

Un primer cas de consulta és sobre documents textuais (textos, pàgines web, llibres, text en bases de dades,...) mitjançant un paraula que es llença sobre el text (consulta lliure com a Google) o una combinació booleana de paraules que es llença sobre camps indexats i controlats (qualsevol tipus de consulta de llibreries digitals, per buscar per autor o títol del llibre).

⁵ www.youtube.com

⁶ www.flickr.com

⁷ www.delicious.com

⁸ www.mpeg.org

Els descriptors en aquests tipus de consultes són també textuals. Alguns exemples de descriptors de baix nivell són la mateixa paraula per a formular una clàssica equivalència caràcter a caràcter o també podem trobar valors estadístics i de recurrència de la paraula de consulta en un text. Descriptors de nivell mitjà podrien ser la llengua de la paraules sobre les llengües dels documents a cercar i descriptors més avançats podrien ser el pes de la paraula dins d'una ontologia semàntica codificada en RDF o OWL. Exemples coneguts són els cercadors web de caràcter general com **Google**⁹ o **Yahoo**¹⁰.

La resposta sol ser un extracte del document o un conjunt d'identificadors dels documents que s'han trobat. El principal problema de les consultes lliures textuals és la ambigüitat dels resultats obtinguts, normalment es mostren molts resultats, però pocs són d'interès.

Descripció	
Consulta	Textual
Cerca	Textuals
Resultats	Visuals

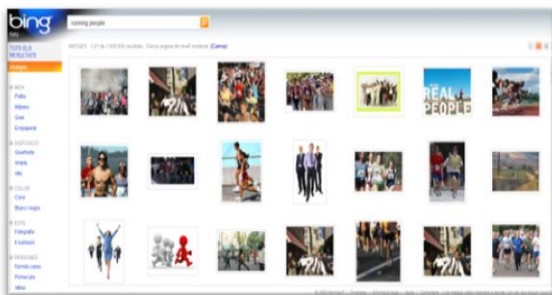


FIGURA 3 MICROSOFT BING

Un segon cas és quan la font de consulta és textual sobre una base de dades d'imatges amb descriptors textuals. Alguns exemples són **Microsoft Bing Images**¹¹ (Figura 3) o **Google Images**¹² (Figura 4). Apart del mateix nom de la imatge o l'extensió del fitxer, trobem descriptors de context segons si la

paraula de consulta s'ha trobat al títol de la web, al cos o tan sols a la direcció URL.

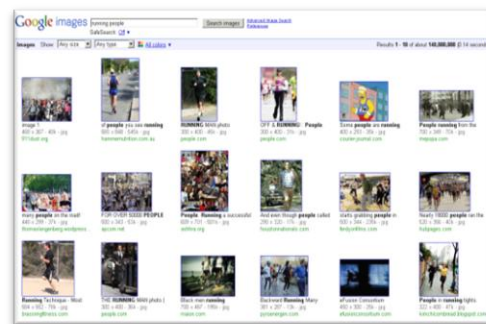


FIGURA 4 GOOGLE IMAGES

⁹ www.google.com

¹⁰ www.yahoo.com

¹¹ www.bing.com

¹² Images.google.com

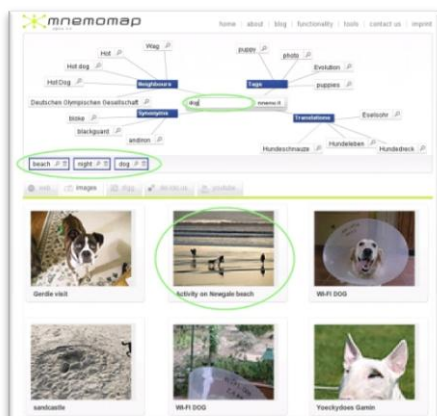


FIGURA 5 MNEMOMAP

Mnemomap¹³ (Figura 5) se centra en oferir recomanacions a la paraula de consulta inicial. Es tracta d'un exemple d'extensió de la consulta a nivell textual. L'eina dona diferents subconjunts de paraules ja siguin sinònimes o bé traduccions a altres idiomes o bé paraules pertanyents a un subconjunt semblant. L'eina permet enllaçar cerques utilitzant la concatenació de paraules.

Un exemple diferent són aquelles eines de consulta sobre imatges que contenen text. Aquest text s'ha extret prèviament amb el que es coneix com a tècniques de OCR. Una eina acadèmica d'aquest tipus és el **Document Image Retrieval System with Word Recognition**¹⁴ (Figura 6).



FIGURA 6 DOC. IMG. RETRIEVAL SYSTEM

Descripció	
Consulta	Visual
Cerca	Visuals
Resultats	Visuals

Aquest tipus d'eines permeten consultes visuals contra bases de dades de paràmetres visuals com el color o la forma.

És el cas cas d'eines com **Multicolor**¹⁵ (Figura 7) on podem entrar una combinació de fins a vuit colors diferents mitjançant una paleta de colors. La consulta es realitza sobre un cnojunt de fotografies seleccionades de Flickr.



FIGURA 7 MULTICOLR

¹³ www.mnemo.org/

¹⁴ orpheus.ee.duth.gr/irs2_5/

¹⁵ labs.ideeinc.com/multicolor/

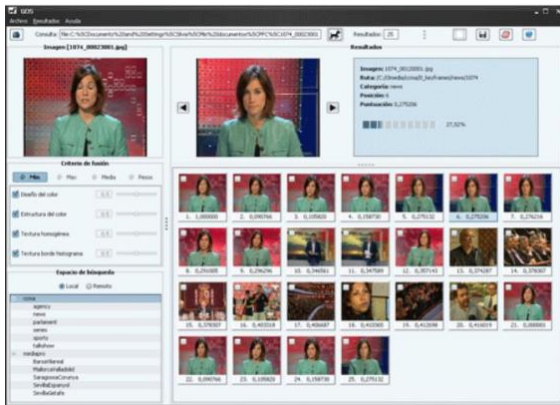


FIGURA 8 G.O.S.

Molt més desenvolupades són eines com l'aplicació **GOS (Graphical Object Search)**¹⁶ [Cortes]^{vii} (Figura 8) del departament de processat de senyal de la Universitat Politècnica de Catalunya, permet fer consultes d'un conjunt de fotogrames extrets de vídeos de producció de TV3 a partir d'una imatge d'un fotograma o d'una regió d'aquest fotograma.



FIGURA 9 IMGSEEK

Un altre exemple acadèmic és **ImgSeek**¹⁷ (Figura 9), una eina basada en la descomposició *wavelet* de les imatges, que en crea una representació única i simplificada. Donada una imatge o un esbós com a entrada, es processa i es fa una consulta de la seva transformada sobre aquesta base de dades.

Simplicity¹⁸ [James et al.]^{viii} (Figura 10), és una eina molt semblant al GOS, permetent la consulta fent un sol clic sobre la imatge per buscar-ne similituds.

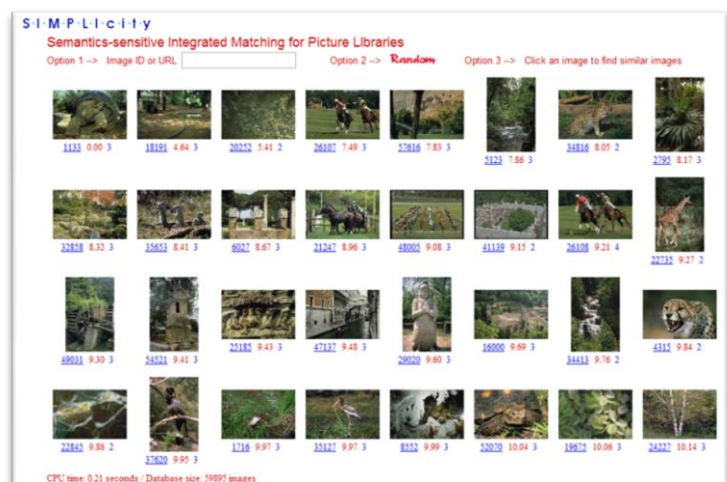


FIGURA 10 SIMPLICITY

¹⁶ gps-tsc.upc.es/imatge/i3media/gos/

¹⁷ www.imgseek.net/

¹⁸ wang14.ist.psu.edu/cgi-bin/zwang/regionsearch_show.cgi

Descripció	
Consulta	Textual+Visual
Cerca	Visuals
Resultats	Visuals

Aquestes eines sorgeixen degut a la dificultat en la avaluació de la semblança entre descriptors visuals, un procés computacionalment molt costós. Per això sovint es realitza primer una cerca textual que redueix el conjunt de cerca i permet un posterior filtrat emprant descriptors visuals.

Una aplicació a nivell de desenvolupament és el sistema **cuZero [cuZero]**^{19 ix} (Figura 11) de la universitat de Columbia. El cuZero es centra més en la disposició de resultats a partir d'una cerca textual que conté més d'una paraula. El sistema proposa l'ús de dos quadrícules, en una

es presenten les imatges recomanades i en l'altre es crea un mapa a on l'usuari pot situar-se segons vulgui donar més pes a un a de les paraules o a una de les imatges. Les imatges van canviant segons la posició del cursor en el mapa. La retícula assigna pesos a les diverses consultes textuals i a les característiques visuals. Depenent de a on es posiciona el cursor en una es visualitzen un nous resultats combinant els resultats prèviament calculats per cada concepte. L'investigador ha testejat altres disposicions però es decanta pel sistema reticular. Aquest sistema no necessita aprenentatge previ per part dels usuaris al ser al ser una visualització molt utilitzada.

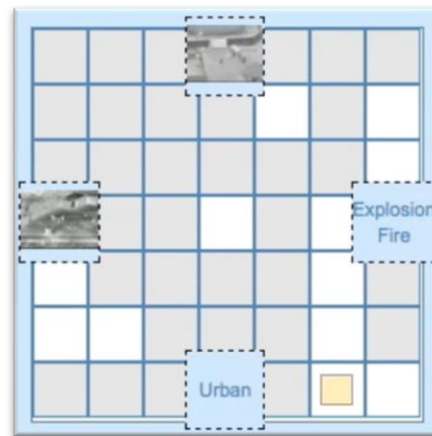


FIGURA 11 CUZERO

¹⁹ <http://www.ee.columbia.edu/ln/dvmm/researchProjects/MultimediaIndexing/cuzero/>

Behold (Figura 12) és una eina bàsicament textual sobre les paraules etiquetades a les fotos de Flickr. Però aquesta consulta està acompanyada d'una llista d'opcions que respon a la pregunta "a què s'assembla la foto que busques?" en anglès *look like* i que internament treballa sobre descriptors visuals extrets dels píxels de les imatges. Un exemple

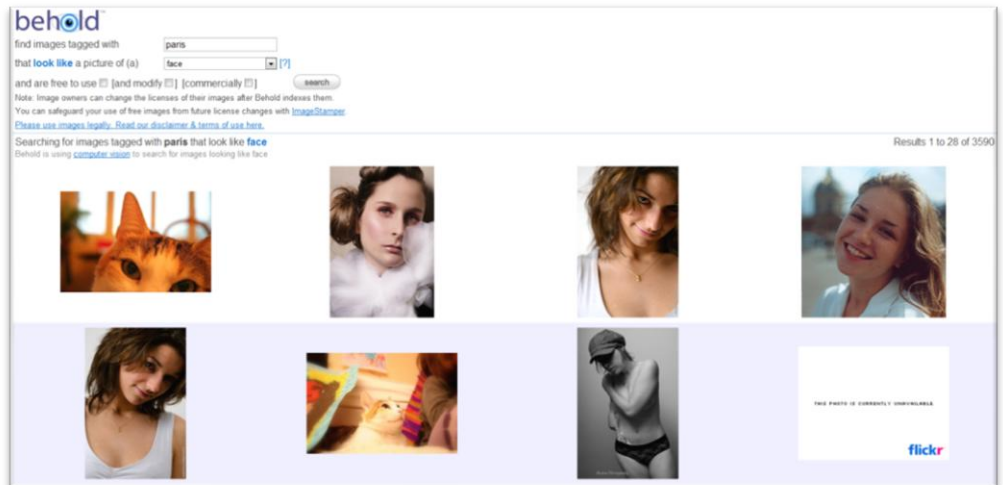


FIGURA 12 BEHOLD

clarificador és buscar la paraula **paris** amb un *look like* de **cares**.

Google Similar Images (Figura 13) és una evolució de l'anterior eina d'aquesta companyia que permet com a entrada una imatge localitzada prèviament sobre una consulta textual. A partir dels primers resultats, alguns d'ells permetran clicar sobre el botó "Similar Images" per a un tipus de consulta visual.

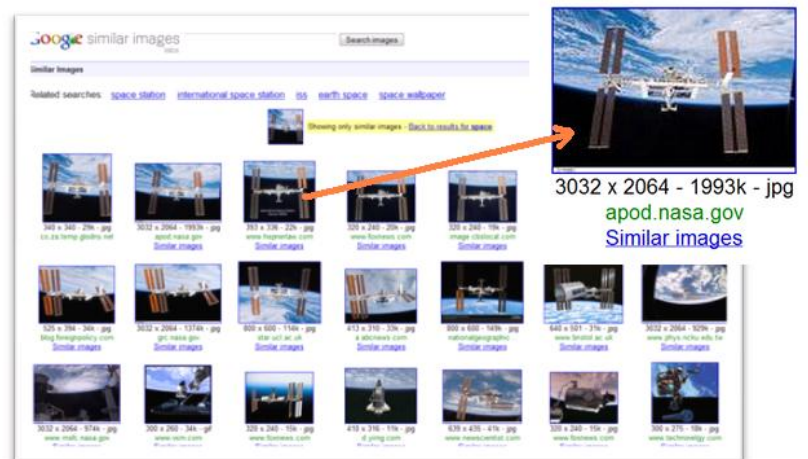


FIGURA 13 GOOGLE SIMILAR IMAGES

Google Swirl (Figura 14) utilitza la mateixa tecnologia que Google Similar Images, però presenta els resultats de forma diferent. Un cop entrada una paraula al quadre de text, Google presenta les imatges resultat agrupades dins d'una circumferència segons la similitud de les seves característiques visuals. Cada nou clic a una imatge proporciona una nova circumferència amb una agrupació lleugerament diferent a l'anterior.

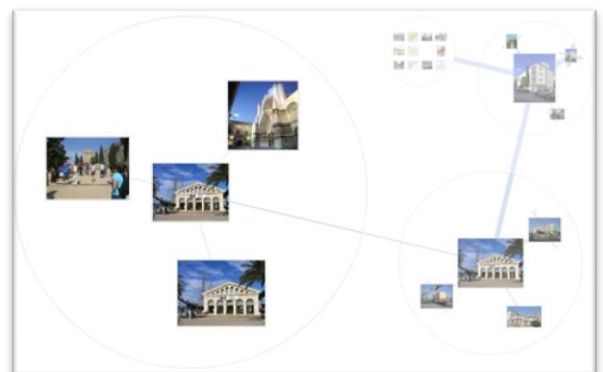


FIGURA 14 GOOGLE SWIRL

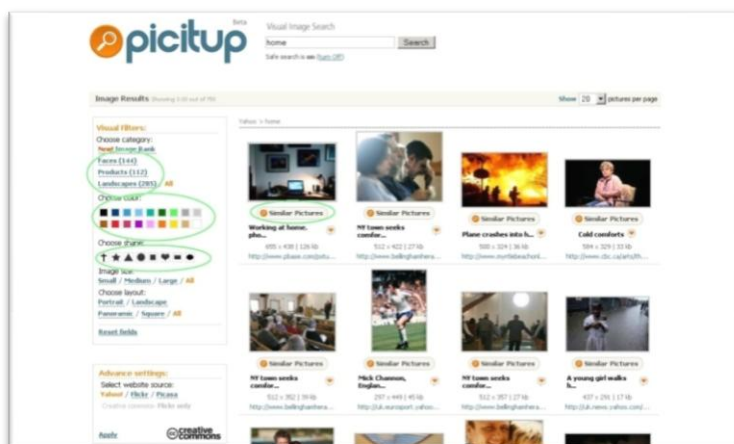


FIGURA 15 PICITUP

Similar és l'eina **Picitup**²⁰ (Figura 15), una tenda de compres. Té un mòdul que permet seleccionar a més del color, la forma de la nostra cerca (una creu, una estrella, un triangle...).

L'eina **Mirror**²¹ (Figura 16) especifica clarament sobre quins descriptors numèrics treballa. Específicament utilitza quatre descriptors MPEG-7 de color i dos descriptors MPEG-7 de textura. L'eina utilitza la retroalimentació de l'usuari per recalculer la mesura de semblança numèrica per a detectar encerts en la consulta.

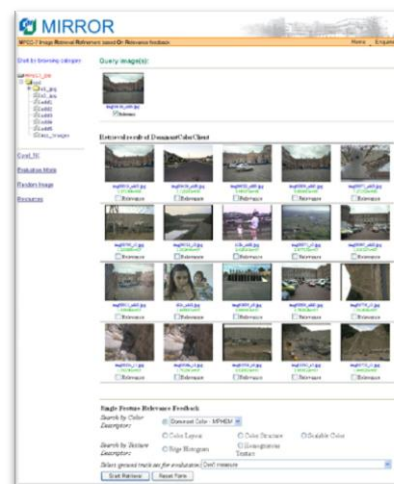


FIGURA 16 MIRROR

Sapir²² [Kaplan et al.]^x (Figura 17) és un projecte liderat per diferents empreses i institucions com IBM o Telefónica que ha desenvolupat una eina amb el mateix funcionament que Google Images. L'eina emprà la mateixa idea d'un hipervincle amb la paraula "similar" per a iniciar les cerques basades en imatge.

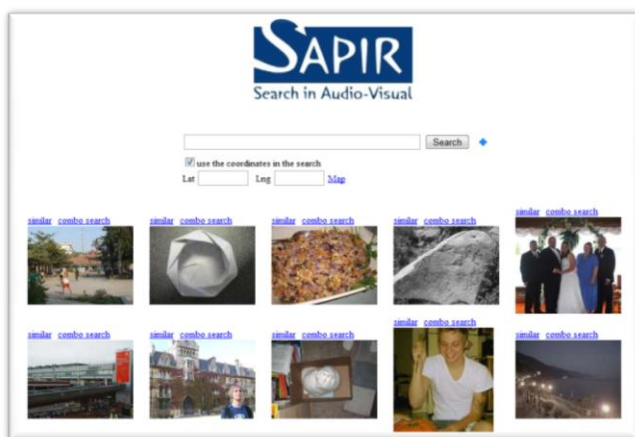


FIGURA 17 SAPIR

²⁰ www.picitup.com/picitup/picShop.jsp

²¹ abacus.ee.cityu.edu.hk/~mpeg7/

²² www.sapir.eu/

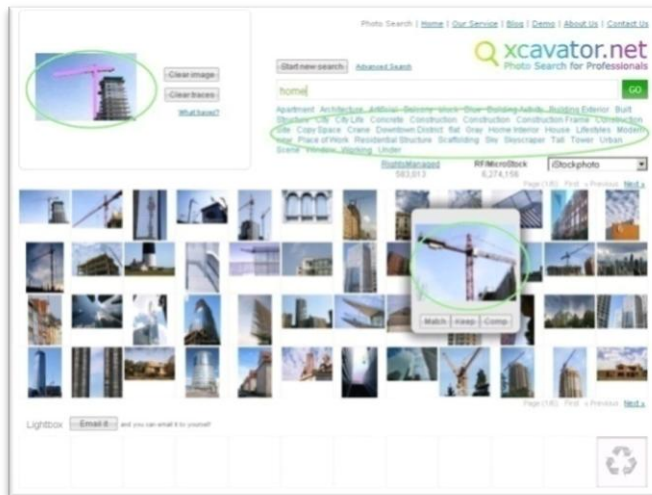


FIGURA 18 XCAVATOR

També trobem **Xcavator**²³ (Figura 18), una eina de resultats molt acurats sobre bases de dades de fotografia professional. L'eina permet ressaltar mitjançant línies, les zones més rellevants de la imatge i iniciar així la consulta. Cal ressaltar el núvol de paraules similars que recomana com a suport per una cerca més acurada.

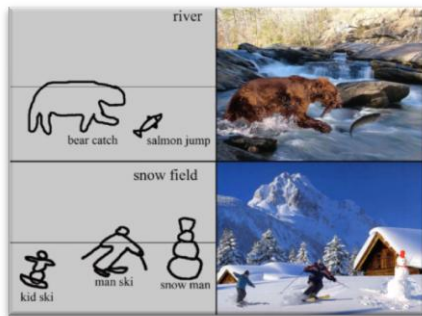


FIGURA 19 SKETCH2PHOTO

L'eina **Sketch2Photo**²⁴ (Figura 19) permet la consulta de fotos a partir de duples esbós més paraula. Les figures que apareixen a l'esbós es dibuixen mantenint una línia de terra que ajuda al programari a fer un primer filtrat sobre una consulta textual. Després s'analitzen els contorns de les figures de l'esbós per determinar quines fotografies són més rellevants.

Pixolu²⁵ (Figura 21) és una eina visualment molt efectiva. S'inicia amb una consulta textual (en la imatge és *tomaques*) i el número de imatges total a carregar. Aquestes imatges es posicionen sobre un mantell del qual es restringeix el número màxim que s'ensenyà utilitzant una màscara que es pot engrandir o empetitir. A partir d'aquesta visualització, es permet indicar una o més imatges semblants a la que es busca per tornar a llençar una consulta redefinida.

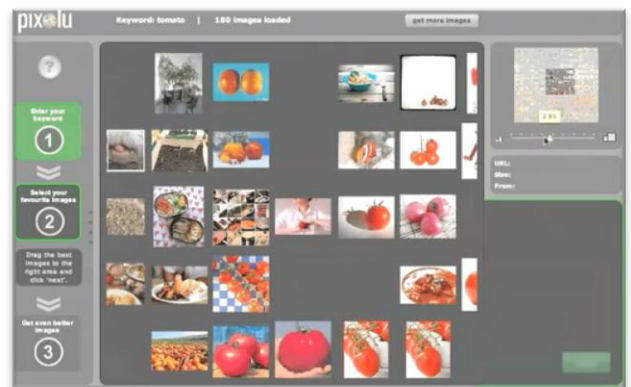


FIGURA 20 PIXOLU

²³ www.xcavator.net/

²⁴ cg.cs.tsinghua.edu.cn/montage/main.htm

²⁵ www.pixolution.de/Pixolu_en.html

Descripció	
Consulta	Visual
Cerca	Textual
Resultats	Textuals

Aquest cas cal ser introduït més acuradament per ser menys comú que els altres i menys desenvolupat. Una eina dins d'aquesta categoria simplificaria aquelles consultes sobre bases de dades en les que a vegades és més senzill expressar la idea a cercar mitjançant una imatge que no una descripció textual. Suposant el cas que es tingui una fotografia d'un acte polític d'un any concret però no es sabés en quin context va ocórrer, la consulta utilitzaria els patrons reconeguts com el reconeixement de les cares de les persones que hi apareixen (s'extraurien descriptors textuals com els noms), els colors de la roba que porten (s'extraurien descriptors textuals), la sala o exterior a on es troben (descriptors visuals), etc... i amb aquest conjunt de descriptors es podria trobar a la Wikipedia, per exemple, informació de la fotografia.

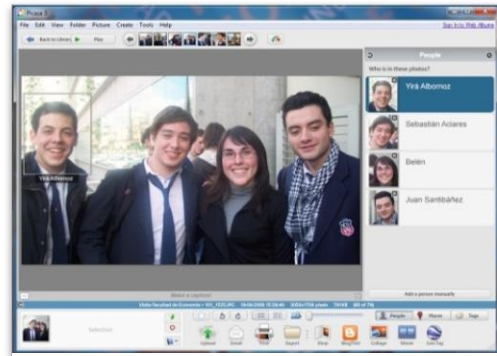


FIGURA 21 GOOGLE PICASA

Com a pas previ, aquests sistemes de consulta necessiten l'**anotació automàtica** de les imatges amb etiquetes textuals. Aquestes etiquetes permetent posteriors **cerques textuals**.

Així doncs, dins l'estat de l'art d'eines d'auto-etiquetat trobem aplicacions de

reconeixement facial com **Google Picasa**²⁶ (Figura 22) que permeten entrar una fotografia per a etiquetar-hi automàticament les persones que hi apareixen i permeten després la seva consulta o bé pel nom de la persona o bé per una imatge de la persona prèviament etiquetada.



FIGURA 22 G.A.T.

L'equip de Processat de la Imatge de la UPC està desenvolupant el **GAT**²⁷ (**Graphics Annotation Tool**) [Giro-i-Nieto et al.]^{xi} (Figura 23) que permet navegar sobre

²⁶ picasa.google.com/

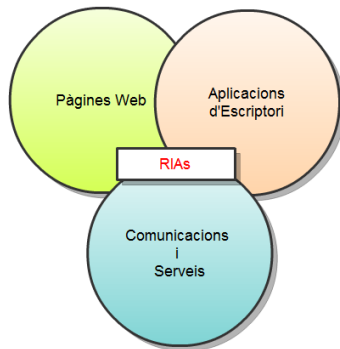
²⁷ gps-tsc.upc.es/imatge/i3media/gat/

subconjunts de píxels rellevants d'una imatge i anotar els possibles objectes que s'hi trobin. Aquesta informació permet la caracterització de models basats en regions que serviran com a input per a fer consultes semàntiques sobre eines de consulta o detecció.

5.2 INTERFÍCIES WEB ENRIQUIDES

5.2.1 APLICACIONS RIQUES D'INTERNET I SERVEIS WEB

Des de ja als principis dels anys 90 amb la primera tenda reeixida d'Amazon fins l'actual ventall de possibilitats que ofereix la web (compra de tiquets, tendes online, gestions administratives, banca...) les pàgines web han anat evolucionant en recursos, serveis, accessibilitat, interoperabilitat i disseny.



Per altra banda, les aplicacions d'escriptori cada cop necessiten beure de més fonts d'informació distribuïdes en xarxa per a ser competitives i donar valor afegit a l'usuari que les utilitza. Avui en dia ja no s'entenen els reproductors d'àudio sense inclusió de podcasts, o lectors de notícies sense accés al format de sindicació RSS, o tampoc els diccionaris sense actualitzacions de la web ni editors de fotografies que no permetin el seu arxivament online.

Des dels primers passos d'Eudora Mail fins a Gmail han passat uns anys on les possibilitats de mobilitat i usabilitat s'han vist disparades pels les connexions de banda ampla i per la proliferació de dispositius com els telèfons mòbils, els ultraportàtils, o el mateix televisor.

És lògic, doncs, pensar que aquests dos tipus d'accés a la informació, ja sigui mitjançant una web pura o un programa d'escriptori pur, estan obligats a entendre's i a enriquir-se mútuament. Aquest és el paradigma de les anomenades Aplicacions Web Enriquides.

Tanmateix, cada cop es busca més immediatesa en l'accés a les dades de l'usuari, indiferentment d'on estiguin, seguint el concepte actual: una sola pantalla amb múltiples serveis, un sol servei amb múltiples pantalles.

Aquesta necessitat passa per *penjar* totes aquestes dades personals, aplicacions o part d'elles a servidors remots indiferentment de si aquests servidors estan a l'habitació del costat o a una habitació a 15º a una empresa de la Xina. De fet, podria ser que una fotografia d'un usuari de l'estiu del 2001 estigui dividida en 50 grups binaris cadascun allotjat en un servidor diferent. Per vertiginós que això sembli, el grau de transparència assolit a dia d'avui permet generalitzar aquesta globalització de dades en un concepte ben senzill: podríem dir que tota la informació generada pels usuaris i el seu processament estan



penjats als núvols. Neixen així els conceptes de Servei Web i de Càlcul als Núvols (*Cloud Computing*).

5.2.2 CÀLCUL ALS NÚVOLS A L'EMPRESA

Com ja s'ha avançat en la introducció, el concepte de **Càlcul als Núvols** neix de la necessitat d'orientar (evolucionar) les necessitats de l'usuari(aplicacions) cap a un model d'informació (dades) descentralitzat.

Com ja va passar amb el concepte de **Web 2.0**, el concepte *Cloud* pot tenir moltes accepcions diferents depenent del context. Però per norma general, el terme càlcul als núvols està altament lligat a totes aquelles preguntes clàssiques de sistemes: Quin espai de disc necessito per incloure-hi les meves dades. Quin tipus de servidor es necessita. Pot suportar accés de cinc usuaris a un mateix objecte audiovisual? I si es vol que el servidor sigui escalable per a un accés de més de mil usuaris? Què passa si es divideix un arxiu en diferents parts? Dos servidors diferents son interoperables? És segur l'accés al document que hi ha al servidor? Es multipliquen dos números al servidor i es passa el resultat al client? O s'envien els dos valors al client i que ell els multipliqui...

La pregunta està clara, com pot ser que un terme tan de baix nivell sigui un dels termes més en boga²⁸ actualment (Figura 24)? La resposta passa per el creixent interès de les empreses en els serveis localitzats a Internet. Moure un model d'empresa als núvols implica fer balança d'uns riscos elevats que, acceptats en mesura, poden aportar beneficis importants. El reportatge d'*InfoWorld*²⁹ citat per moltes revistes especialitzades dóna les claus de l'acceptació en l'empresa:

La Computació als Núvols entra en joc que es pensa en allò que un departament de IT sempre necessita: una forma d'incrementar o afegir noves capacitats en calent (on the fly) sense haver d'invertir en una nova infraestructura, entrenament de nou personal, o llicències de nou programari. Una subscripció a una empresa de Cloud Computing

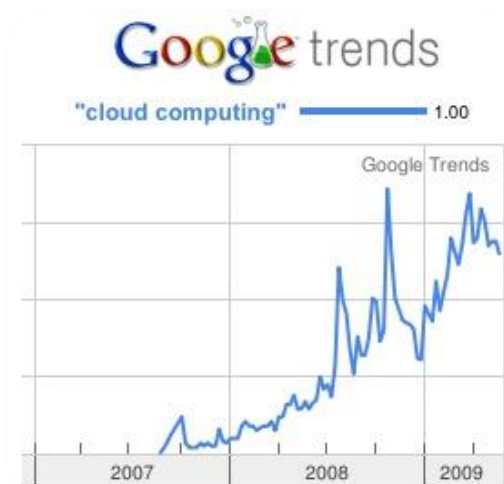


FIGURA 23 RELLEVÀNCIA DEL CLOUD COMPUTING

²⁸ blogs.zdnet.com/Hinchcliffe/?p=488

²⁹ www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031

genera els beneficis de deslliurar-se de problemes d'escalabilitat a preus molt més accessibles que els generats pel fet de mantenir tota una infraestructura en local.

El Càlcul als Núvols avarca moltes solucions (fins i tot filosofies de treball) diferents per a l'empresa, es descriuen a continuació les tres més interessant que es desenvoluparan al llarg d'aquest projecte:

SAAS (SOFTWARE AS A SERVICE)

És el tipus de Càlcul als Núvols en què es centra aquest treball. Tracta de disposar d'una aplicació que es mostra mitjançant un navegador i es serveix a centenars d'usuaris mitjançant una arquitectura distribuïda. El principal avantatge és l'alliberament de les necessitats de manteniment de programari en local en una empresa. Altres variants depenen de la seva complexitat són: **IaaS (Infraestructura as a Service)** i **PaaS (Platform as a Service)** (Figura 25).

UTILITY COMPUTING

És el concepte de manteniment de recursos no imprescindibles als núvols però que poden ser susceptibles de créixer en un futur immediat. Com per exemple, contractar espai en disc *online* o contractar capacitat de processament en servidors aliens... L'avantatge principal és la elasticitat del servei.

WEB SERVICES IN THE CLOUD

És un concepte altament relacionat amb el **SaaS**, i dóna la possibilitat de generar contractes per l'ús puntual o per subscripció a dades *online*. Com per exemple, contractar un servei de consulta d'estocks *online*, consultar informació mitjançant APIs obertes com Google Maps³⁰ o Flickr³¹, o bé establir un model de mercat basat en

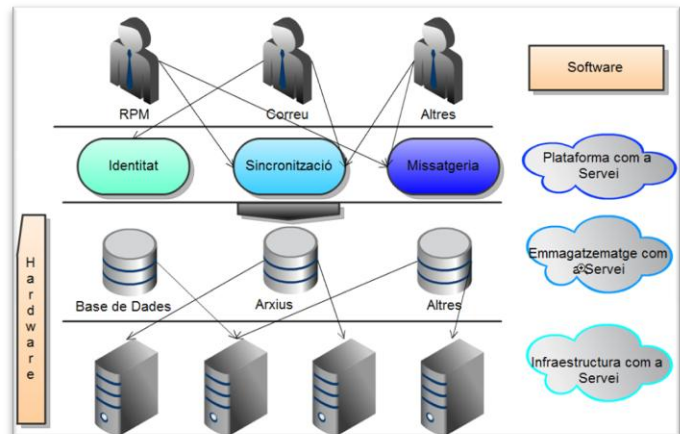


FIGURA 24 VARIANTS DELS SERVEIS WEB

pagament per accés a un conjunt de dades. Podríem dir que aquesta solució és un SaaS parcial però que es regeix segons un patró propi molt centrat a les dades i no tant en com s'utilitzaran a posteriori aquestes dades.

³⁰ Maps.google.es/maps

³¹ www.flickr.com



FIGURA 25 ONLIVE SERVICE

parts de fitxers, o la recent incorporació al mercat de les videoconsoles de el sistema onLive³² (Figura 26). En aquest darrera cas, es processen les ordres del comandament de jocs d'un usuari en un servidor remot que renderitza el vídeo i el retorna cap al client fent que per primer cop, una consola no disposi ni de processador ni de targeta gràfica.

Actualment, a nivell de mercat, alguns dels proveïdors més reconeguts de serveis *Cloud* són Amazon Elastic Compute Cloud (Amazon EC2)³³ que permet córrer als núvols una distribució Linux amb una base de dades i un servidor web amb APIs per al seu manteniment i Accés. Google App Engine³⁴ i Windows Azure³⁵ en són els seus directes competidors. Aquests proveïdors proveeixen de solucions tancades i assequibles per a córrer ja sigui tota una aplicació sobre un servidor remot o només una API que doni accés a una taula de dades, és a dir, donant suport a la computació als núvols en qualsevol de les seves vessants.

També es troben exemples més sofisticats del concepte de *Cloud Computing* com els que pretenen fer computació distribuïda a través de

xarxes d'ordinadors, la descàrrega P2P mitjançant distribució anònima de

³² <http://www.onlive.com>

³³ <http://aws.amazon.com/ec2/>

³⁴ <http://code.google.com/intl/es-ES/appengine/>

³⁵ <http://www.microsoft.com/azure/>

5.2.3 DEFINICIÓ DEL CONCEPTE D'APLICACIÓ WEB ENRIQUIDA

Avançant en el projecte, ens centrem ara en definir aquelles aplicacions que tenen per objectiu cobrir una necessitat d'usuari i que es poden incloure en el concepte definit anteriorment com a **SaaS** (*Software as a Service*).

El concepte de d'Aplicació Web enriquida (*RIA - Rich Internet Application*) neix l'any 2002 a l'empresa Macromedia en el document³⁶ anomenat "Macromedia Flash MX—A next-generation rich client" on es presentava un nou producte de la companyia³⁷.

La indústria ràpidament va fer propi el concepte per a definir aquelles aplicacions programades que segueixen quatre requisits bàsics³⁸:

- ✓ El funcionament del programa sobre un motor d'execució propi.
- ✓ La integració de Contingut, Comunicacions i Interfícies d'aplicació en una experiència comuna.
- ✓ Disposició d'una arquitectura orientada a serveis que utilitzi les dades proveïdes per servidors.
- ✓ La seva implementació independent de la plataforma, sistema operatiu, navegador web i dispositiu on funcionés.

Tanmateix, el concepte ha anat evolucionant des d'una perspectiva de programació cap un nou concepte de facilitat d'ús en les aplicacions web, tot maximitzant la seva principal avantatge: la experiència d'usuari. Per aquest motiu es poden trobar diferents noms aplicats a aquest concepte d'usabilitat: aplicacions enriquides, webs enriquides, webs dinàmiques, aplicacions orientades a serveis, aplicacions *stateful* (que mantenen l'estat)...

La nova generació de serveis en línia pretén corregir algunes limitacions de les pàgines web tradicionals que es limiten a renderitzar informació en format **HTML**. Per exemple, en un procés de compra de bitllets d'avió sovint l'usuari ha d'omplir un llarg formulari, fer un clic al botó d'acceptació i carregar una nova plana abans de saber si hi ha un error en les seves dades. Si és així, haurà de fer clic a la fletxa del navegador per a tornar enrere, tornar a omplir totes les dades, fer clic al botó d'acceptació i potser es trobarà que ha acabat reservant dos cops el mateix vol. Aquesta limitació pot portar en el pitjor dels casos que l'usuari decideixi no fer la reserva del vol o fins i tot valorar negativament l'empresa que ofereix el servei.

³⁶ www.adobe.com/devnet/flash/whitepapers/richclient.pdf

³⁷ www.adobe.com

³⁸ www.adobe.com/devnet/logged_in/thale_rias.html

5.2.4 DIAGRAMES COMPARATIUS

Per entendre el nou concepte d'experiència cal fer un procés de replanificació dels passos que s'executen des que un usuari inicia un servei fins que l'acaba, enlloc de programar una web exclusivament pensant en cada una de les accions que l'usuari pren. Els següents diagrames comparen els fluxos d'ús de les webs tradicionals amb les aplicacions riques:

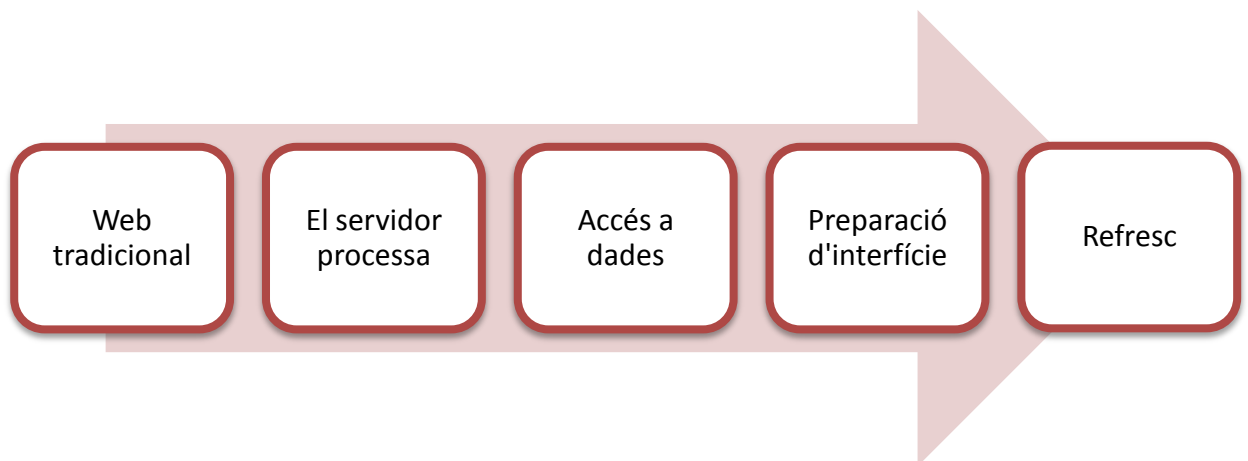


FIGURA 26 PROCÉS DE WEB TRADICIONAL

En el primer cas (Figura 27), ens trobem amb una estructura determinada per la resposta a cada acció de l'usuari. L'usuari fa clic a un botó en una web presentada per un navegador. Aquest servidor processa la petició tot agafant les dades d'alguna font accessible. El servidor agafa alguna plantilla o elements d'interfície (HTML, CSS, imatges, etc...) per a mostrar a l'usuari les dades que ha trobat. El servidor finalment refresca les dades i la interfície per tal que l'usuari visualitzi la informació.

Cal doncs fer crides al servidor per a cada pas que es dona en l'ús de la web. La interfície està lligada a aquestes crides de tal forma que innecessàriament cal refrescar la plana per a cada clic que l'usuari fa.

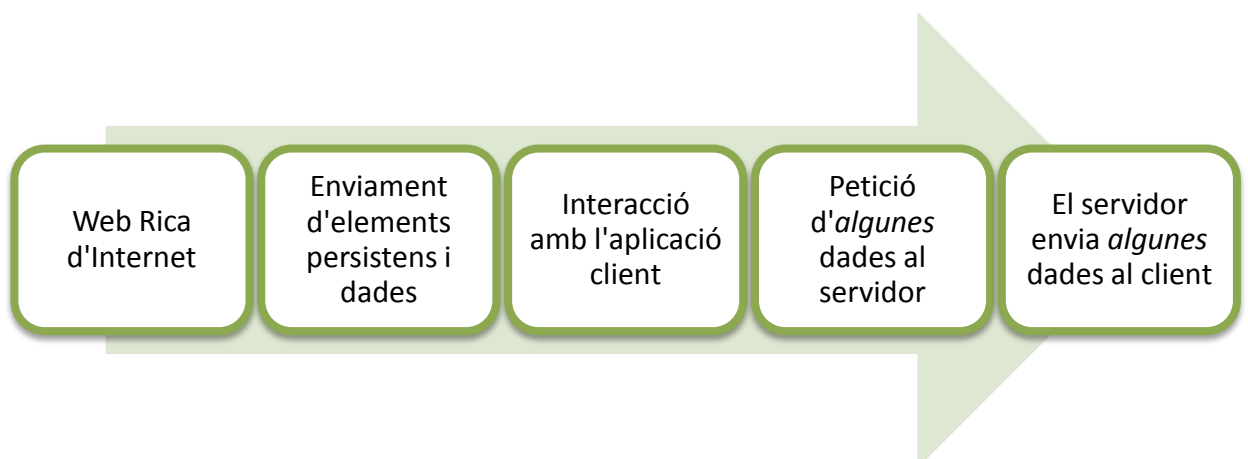


FIGURA 27 PROCÉS D'APLICACIÓ WEB ENRIQUIDA

En el segon cas (Figura 28), l'aplicació ha estat pensada per permetre a l'usuari preparar les dades abans de notificar cap canvi cap al servidor. Això és possible gràcies als elements gràfics persistents que no varien durant el transcurs de l'operació, és a dir, botons, taules, gràfics, elements bàsics de qualsevol formulari són carregats prèviament i un sol cop al pla client. L'usuari pot interactuar amb ells sense necessitat d'informar al servidor que hi està treballant. Un altre punt diferenciador és que mantenir elements persistents al llarg de l'ús de l'aplicació fa que el nombre de dades en primera càrrega sigui major, però un cop carregades, les successives crides al servidor són més lleugeres. És a dir, es prescindeix del refresc de la interfície per a centrar-se en el moviment de les dades.

En resum, el principal puntal de les **RIAs** (i un dels trets bàsics per a identificar-les) és el fet de deslligar-se de la recàrrega plana a plana per a respondre a cada acció de l'usuari.

El concepte d'usabilitat no és baladí. A [Luttinger]^{xii} podem llegir que existeix una correlació clara i directa entre, per una banda, la facilitat i l'enriquiment de l'experiència en l'ús d'un programa, i per l'altra banda, les peticions de modernització dels programaris antics d'una empresa per part dels usuaris:

Els treballadors d'una empresa volen buscar entre els seus documents tal i com ho fan amb els cercadors web, volen reproduir vídeos sense dependre de codecs (codificadors i descodificadors), volen reiniciar l'ordinador i trobar-se el programa en l'últim estat on l'han deixat (persistència) com fan amb els serveis personalitzats iGoogle³⁹ o Netvibes⁴⁰. Volen moure's per tota l'empresa sense dependre del terminal a on es connectin. Volen evitar fer tres clics per a una tasca repetitiva si només en cal fer un.

Totes aquestes peticions poden ser ateses per aplicacions RIA ben dissenyades.

5.2.5 AVANTATGES I INCONVENIENTS DE LES RIAs

Les principals avantatges de les aplicacions web enriquides són:

- ✓ La rapidesa d'accés.
- ✓ La flexibilitat en l'ús.
- ✓ La interactivitat entre un o més usuaris.
- ✓ El disseny atractiu que ofereixen gràcies a la fàcil inclusió d'elements multimèdia.

³⁹ www.google.es/ig

⁴⁰ www.netvibes.com

- ✓ Una clara diferenciació entre interfície i servidor, dividint les tasques del programador de les tasques del dissenyador gràfic.
- ✓ Són independents del sistema operatiu en el que s'executen i del navegador escollit.
- ✓ Són aplicacions lleugeres i no requereixen instal·lació.
- ✓ Són de fàcil manteniment i els canvis són ràpidament estesos als usuaris.

D'altra banda, com a inconvenients podem trobar:

- ✓ La necessitat d'un servidor més complex que un servidor web tradicional.
- ✓ Correlació entre el bon funcionament de l'aplicació i l'ampla de banda de què disposi l'usuari.
- ✓ Són més complicades de dissenyar, desenvolupar i detectar errors que aplicacions tradicionals.
- ✓ La centralització de recursos fa que la caiguda del servidor afecti a tots els clients que hi depenen.

5.2.6 ENTORNS DE PROGRAMACIÓ DE RIAs (*FRAMEWORKS*)

Una cop explicades la filosofia i els avantatges de programar una RIA, cerquem el conjunt d'eines de què disposa el programador per a la planificació, la programació, la cerca d'errors, la implementació i l'execució d'una RIA.

Aquestes eines, van des de petites llibreries **JavaScript** fins a entorns avançats com Eclipse⁴¹. La seva orientació dependrà de si es vol dissenyar un petit giny (*gadget*) d'escriptori que mostri per exemple, un canal de subscripció (*feed*) de les nostres amistats de Facebook⁴² o tota una aplicació de compra venda mitjançant Ebay⁴³.

Els llenguatges més emprats són AS3, .NET, PHP, XML, CSS, HTML (V5) i Java. Els *frameworks* més emprats són Flex, Silverlight, WebStart basats en AS3, .NET i Java respectivament i qualsevol *framework* que es desenvolupi en JavaScript i Ajax com JQuery, Dojo Toolkit o Google Web Toolkit.

El principal avantatge de programar RIAs en JavaScript i Ajax és que JavaScript està suportat per tots els navegadors web per defecte des de Netscape Navigator 3 l'any 1995. Per altra banda el seu principal problema és estar lluny de la maduresa de solucions més tancades com Flex i Silverlight que disposen cadascuna d'un connector (*plug-in*) únic per a la majoria de navegadors que interpreten els programes. Aquest problema s'intenta resoldre amb la nova versió **HTML 5**. Aquest llenguatge proposa noves etiquetes destinades a fer que una mateixa disposició (*layout*) d'elements sigui

⁴¹ <http://www.eclipse.com>

⁴² <http://www.facebook.com/>

⁴³ <http://desktop.ebay.com/>

reproduïda fidelment per tots els navegadors que l'implementin. Alguns exemples són les noves etiquetes `<canvas>`, `<section>` o `<article>`.

El cas del Java WebStart és una evolució dels anomenats Applets de Java, mini aplicacions que corrien sobre el navegador emprant el plugin JRE de Java. Java ha millorat la tècnica incloent WebStart dins el seu plugin JRE permetent una aplicació més RIA sobre qualsevol aplicació programada en Java, el principal desavantatge és la grandària que ocupen tant el connector JRE (aproximadament 14 MB) com els programes que acaben sent desenvolupats sobre Java.

Específicament, el connector d'Adobe, l'anomenat Flash Player té una penetració⁴⁴ del 98% en els ordinadors d'usuari de tot el món, seguit només pel **Java Runtime Environment**⁴⁵ de l'empresa Sun. També notar que el sistema operatiu Windows 7 de Microsoft duu incorporat el runtime de Silverlight com a avantatge competitiu.

5.2.7 PERSISTÈNCIA DE LES RIAS

Les aplicacions RIA que empen connectors de navegador tenen algunes limitacions derivades del fet de dependre de l'embolcall d'un navegador. És a dir, depenen dels recursos designats al navegador que poden ser inferiors als recursos que requereixen aquestes aplicacions.

Els avantatges⁴⁶ d'emprar el sistema operatiu com a sistema final són les inherents a l'obtenció de més recursos de màquina, accés directe a l'escriptura i lectura de fitxers, integració amb el Sistema Operatiu mitjançant icones en la safata (*tray*) o en l'escriptori (*desktop*) i no menys important la possibilitat de determinar si hi ha o no connexió a Internet. En definitiva, les RIA no estan barallades amb el concepte escriptori, sinó que el complementen i donen al programador més opcions per a utilitzar la connectivitat com a avantatge.

Cal entendre que el pas del navegador cap a l'escriptori ha d'executar-se sempre a arrel d'una acceptació de l'usuari ja que passem el control de seguretat que implementa el navegador cap al control que implementa el sistema operatiu. Degut a les múltiples vulnerabilitats dels sistemes operatius actuals, juntament amb el seu accés a molts més recursos màquina, podem dir que les RIA executades sobre Sistemes Operatius són molt més vulnerables a atacs.

Adobe ha implementat el sistema anomenat AIR que permet instal·lar qualsevol RIA desenvolupada pel navegador en qualsevol sistema operatiu. La persistència en

⁴⁴ www.adobe.com/products/player_census/flashplayer/version_penetration.html

⁴⁵ java.sun.com

⁴⁶ www.adobe.com/products/air/comparison/

l'escriptori que permet Adobe Air també està present en les eines de Microsoft Silverlight i Java, sempre i quan es tinguin instal·lades les rutines de .Net i JRE d'ambdues empreses.

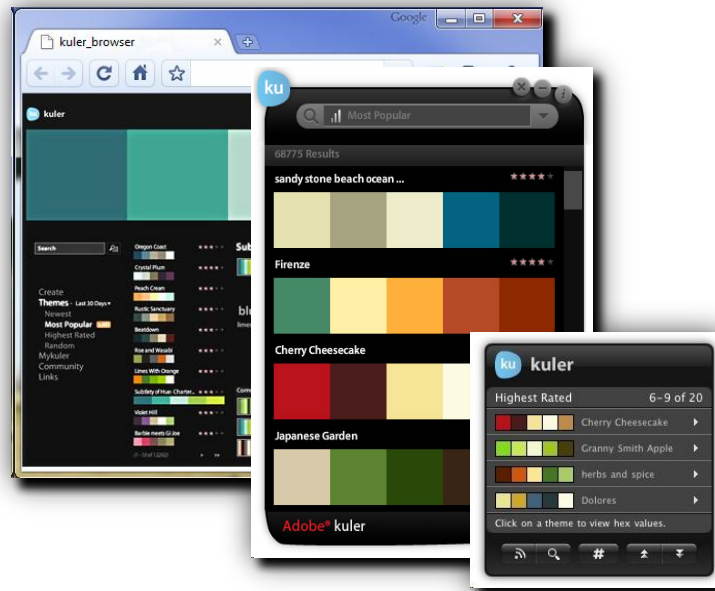


FIGURA 28 ADOBE KULER⁴⁷ EN LES VERSIONS WEB, ESCRIPTORI I GINY

Un exemple de RIAs molt esteses en els sistemes operatius actuals són els ginys d'escriptori que podem trobar en la barra lateral (*SideBar*) de Windows Vista o l'entorn Plasma de KDE. Els ginys solen funcionar en base a consultes a serveis web externs (consulta de la meteorologia, consulta del correu, consulta de *feeds*...). En la Figura 28 es mostra un exemple d'aplicació de disseny en la seva versió web, com a programa d'escriptori o com a giny.

Les RIAs basades en Java Script també estan sent portades als diferents sistemes operatius mitjançant la integració tant en interfícies Linux com Gnome o KDE dels principals motors de renderitzat de JS com són Mozilla TraceMonkey, WebKit SquirrelFish o V8 implementat per Google Chrome. De fet, Google ha anunciat recentment⁴⁸ la seva disposició de crear un Sistema Operatiu basat en el seu navegador i en el seu sistema renderitzat de JavaScript com a màxim exponent d'una aplicació RIA completament lligada als recursos de maquinari i a Internet basada en el *Cloud Computing*.

⁴⁷ kuler.adobe.com

⁴⁸ googleblog.blogspot.com/2009/07/introducing-google-chrome-os.html

5.2.8 SERVIDORS

En l'anterior apartat s'ha parlat dels diferents entorns (*frameworks*) i llenguatges com a alternatives per a la construcció d'aplicacions enriquides. Dins del *workflow* d'aquest tipus d'aplicacions el pas següent a la programació de l'aplicació és l'allotjament de la mateixa en servidors remots ja estiguin en un domini físic accessible o en un entorn més aliè com el que s'ha introduït en el *Cloud Computing*.

En l'actualitat, els principals servidors són Apache i Microsoft ISS tal i com es pot veure en la Figura 29 (font: NetCraft⁴⁹). Apache té llicència de programari lliure i ISS és privatiu.

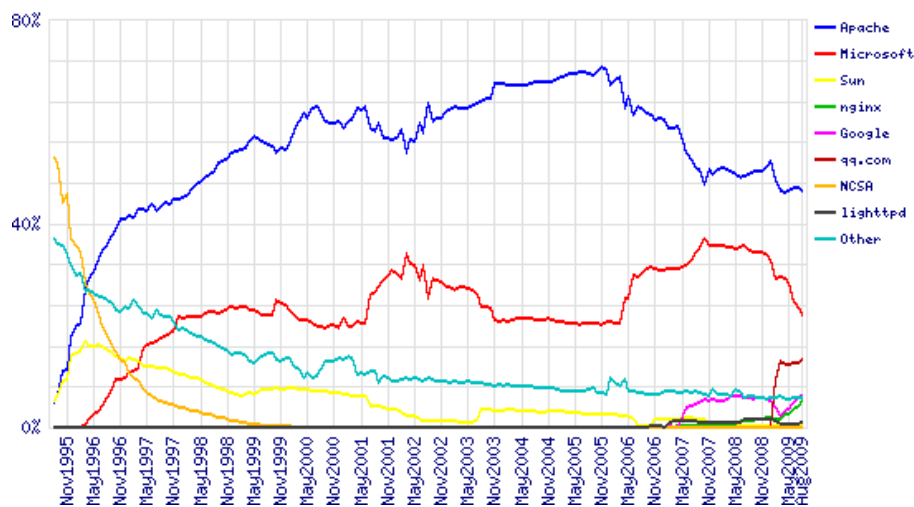


FIGURA 29 ÚS DELS PRINCIPALS SERVIDORS (FONT: NETCRAFT)

Tanmateix, un Servidor Web per si mateix no processa peticions més enllà de les HTTP. Per dotar de més potència, interactivitat i modulariat al servidor, i poder processar les dades rebudes o enviades a les RIAs, cal moure's a llenguatges més solvents i amb **manteniment d'estat** com són Java o .NET o bé llenguatges interpretats com PHP o ASP.

El manteniment d'un estat durant l'execució d'una petició web se sol diferenciar també segons si el llenguatge és compilat o bé interpretat. Així en el cas de Java, les classes compilades han de ser executades i guardades en memòria durant una petició (crida), mentre que un *script* de PHP s'interpreta i executa en memòria a l'instant.

El suport dels principals servidors web a aquests llenguatges es dona mitjançant connectors (*plug-ins*) o mòduls d'ampliació. Alguns connectors coneguts de Java per a l'execució de Servlets són Tomcat o WebOrb.

⁴⁹ news.netcraft.com/archives/web_server_survey.html

5.2.9 PROTOCOLS DE COMUNICACIÓ

La Web 2.0 ha estat impulsora de nous protocols de comunicació i refinament de protocols més consolidats. Les aplicacions RIA mouen conjunts de dades significativament més grans (fitxers media, conjunts exportats directament d'una base de dades, dades de grandària petita però en múltiples i repetides crides...).

Per una banda, trobem protocols que han anat madurant sobre la capa HTTP, com el protocol SOAP o el protocol REST. El primer utilitza fitxers XML per a totes les capes de comunicació, des de la configuració d'un servei fins a l'embolcall de les dades a passar. Com a avantatge es té que s'ha estandarditzat⁵⁰ oficialment per la W3C⁵¹ (consorci web) i s'empra molt en contextos empresarials per oficialitzar (contractualitzar) i fins i tot monetitzar la transferència de dades. Com a desavantatges trobem la sobrecàrrega per *overhead* d'ample de banda que suposa el format XML.

El protocol REST és una simplificació que permet comunicacions més òptimes i menys redundants a canvi d'una pèrdua de rigurositat en la transacció client servidor. REST empra les funcions GET i PUT definides en l'HTML i hi encapsula a l'adreça els paràmetres de crida a transmetre sense l'ús de fitxers XMLs. Les dades poden ser retornades directament en format de text pla, XML o JSON, o en formats estructurats més dirigits a la sindicació de dades com els formats RSS o el format Atom⁵².

Es pot pensar que REST és una tecnologia posterior a SOAP, però de fet, el REST és anterior. Senzillament la tendència actual de prioritzar la simplificació i el menor consum d'ample de banda n'ha facilitat un ressorgiment. Eines Web com Twitter, Flickr, Del.icio.us, Pubsub, Bloglines o Technorati en fan ús exclusiu per a les seves APIs. Altres eines SOAP estan sent depreciades a favor de REST com les APIs SOAP de Google que es mantenen però ja no s'actualitzen.

Segurament el factor que pot fer decaure l'ús de REST en un servei propi formal o empresarial és la seguretat de les dades. Tanmateix, gràcies a l'actual ressorgiment, s'estan compatibilitzant les antigues vies d'encapsulació ja sigui del canal (mitjançant HTTPS) o bé del missatge (WS-Security) per a dotar-ne de més seguretat.

Altres protocols web difereixen dels anteriors per no estar encapsulats sobre el HTTP. És el cas del protocol privatiu RMTP desenvolupat per Adobe i estàndard de facto per a streaming de vídeo i àudio amb suport pel Flash Player de la mateixa empresa. Així, eines com Youtube han impulsat els servidors que suporten aquest protocol com Flash Media Server o l'alternativa de codi obert Red5.

⁵⁰ www.w3.org/TR/soap/

⁵¹ www.w3.org/

⁵² browse.develop.com/2009/json/xml/

Un altre protocol d'Adobe, però en aquest cas de codi obert, és l'AMF (Action Message Format), un protocol de comunicació exclusiu entre servidors Java i clients Flex que permet disposar remotament de les classes Java o de les classes en Action Script 3 mitjançant serialització de dades. L'eina és una avantatge per a aquells programadors que es disposin a treballar amb les dues eines perquè permet mantenir aplicacions **orientades a objecte** utilitzant el mateix objecte indistintament de si es troba en el pla servidor o el pla client. D'altra banda el fet d'obrir el protocol ha aconseguit que aquest comenci a tenir equivalències amb altres llenguatges com .NET (WebORB), PHP (AMFPHP), Python (PyAMF), Perl (AMF:PERL), Curl (Curl Data Services), Ruby (RubyAMF).

6 ESCENARIS D'ÚS I REQUERIMENTS D'USUARI INICIALS

6.1 ESCENARIS D'ÚS

Els usuaris de la Corporació (Corporació Catalana de Mitjans Audiovisuals) necessiten cercar ràpidament vídeos que s'adeqüin als continguts que es pretenen emetre. Els informatius, per exemple, són espais de producció que requereixen d'un temps de resposta molt breu entre que succeeix una notícia i que l'equip productiu acaba creant un vídeo editat amb imatges no només gravades el mateix dia sinó també provinents de l'arxiu. L'accés al repositori es realitza a partir de consultes sobre la seva base de dades d'imatges digitalitzades des de l'any 2001 i en constant augment.

L'equip de documentalistes de la Corporació són els encarregats de processar les consultes dels periodistes, redactors i editors (usuaris professionals) i respondre-les amb vídeos de personatges que són notícia, d'actes rellevants passats, d'edificis i localitzacions a on passa la notícia, etc...

L'actual mètode de consulta passa per l'ús de la capa de cerca textual del **Digiton Digimedia**, un repositori digital creat per la Corporació a on s'ingesten els vídeos, àudios i descripcions dels mateixos. Cada conjunt de vídeo i àudio s'anomena **asset** i un cert nombre d'assets marcats com importants passen per un filtre de l'equip de documentació a on manualment s'anoten i es descriuen minuciosament per tal de no perdre'ls un cop arxivats dins de la seva ingent base de dades.

Aquestes descripcions poden tenir un caràcter global de tot l'asset, com per exemple "*partit de futbol: Barça 3 - Atlètic 1*" o bé poden referir-se a un segment temporal que incorpora codi de temps d'entrada i de sortida de la descripció "*00:05:30 - El president cubà ensopega i cau de l'estrada - 00:06:03*". Anomenarem a partir d'ara aquestes descripcions **estrats**.

Prenent un dia qualsevol, al Digiton hi entren una mitja de 5000 assets dels quals al cap d'un mes han sobreviscut (han sigut marcats per a no ser esborrats) una mitja de 200. Poc menys de dos terços d'aquest 200 assets (120) corresponen a l'arxiu d'informatius i poc menys del terç restant (70) correspon a l'arxiu d'esports, la resta correspon als demés arxius de la Corporació com l'arxiu de produccions audiovisuals pròpies. Finalment, d'aquests 190 assets marcats com a importants, es documenten i es detallen en mitja uns 100 assets d'informatius i uns 60 d'esports. Es pot dir que aproximadament s'anota el 3,2% dels 5000 assets entrats inicialment.

El Digiton Digimedia, és una eina concebuda per ajudar i simplificar el salt digital que va donar la Corporació per substituir el seu sistema de producció basat en cintes de vídeo per arxius digital en tota la cadena d'elaboració d'editats.

A continuació es presenten i es descriuen alguns dels punts visibles que fan al Digition Digimedia subjecte de millora per a la seva nova implementació:

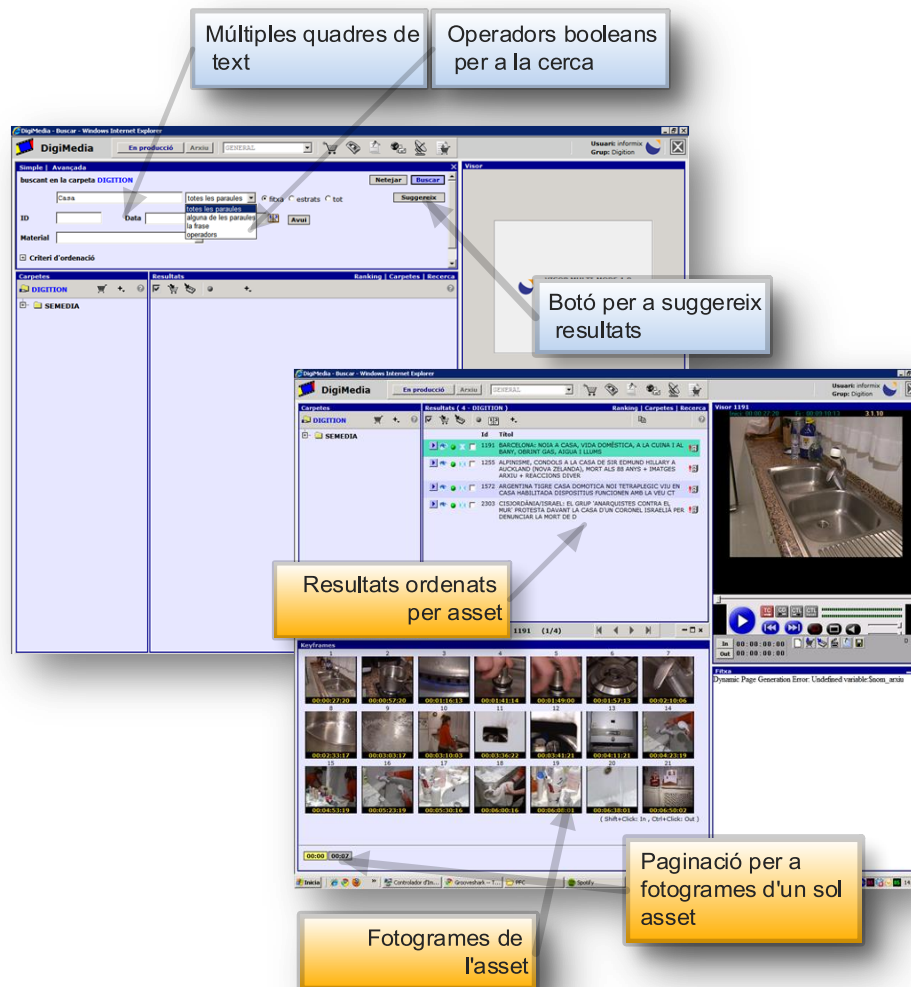


FIGURA 30 DIGITION

La Figura 30 mostra dues captures del programa. De la imatge superior ens interessa el mòdul de cerca on podem observar que està disgregat en diferents quadres de text que permeten atacar diferents camps de la base de dades. Podem observar com també es deixa a l'usuari l'elecció de les opcions booleanes de cerca, fent de la consulta textual un pas enrevesat per a les múltiples combinacions possibles.

A la imatge inferior es poden veure la capa de presentació de fotogrames d'un sol *asset* seleccionat. Els fotogrames estan paginats, és a dir, si l'*asset* té més de 3x7 fotogrames s'haurà d'anar fent clic a les subsegüents pàgines per a mostrar totes les imatges.

Digition avarca una gran quantitat de funcions. Les més interessants per al present treball són:

- ✓ la gestió de la base de dades amb els assets,
- ✓ el sistema d'ingesta d'es d'una càmera de vídeo remota,
- ✓ el sistema de gestió d'usuaris,
- ✓ l'anotació i tractament de metadades,
- ✓ i el manteniment de la seva interfície de cerca.

Aquestes funcions fan del Digiton un candidat excel·lent per:

- ✓ innovar noves maneres de presentar dades,
- ✓ ajudar a l'usuari a trobar allò que cerca
- ✓ i permetre interconnexió amb mòduls avançats de processament que facilitin l'ús diari dels usuaris professionals.

Es van fer un seguit de test d'usuari, sobre treballadors professionals de la casa, per detectar requisits d'usuaris i per tal de complementar i millorar la capa de consulta del Digiton Digimedia. Aquests tests es van emmarcar en el marc de **l'activitat 10 del projecte i3media [i3media]^{xiii}**. L'estudi va dibuixar unes necessitats bàsiques que s'han anat delimitant mitjançant consultes directes sobre els periodistes o documentalistes. Els requeriments del present projecte són fruit d'una extensa interacció amb els futurs usuaris del sistema mitjançant correus electrònics, llegint documents que generen durant la seva feina o fins i tot, durant tests d'usuari específics sobre l'eina que es presenta i que ha anat evolucionant durant la realització dels mateixos.

6.2 REQUISITS DEL SISTEMA

Un cop vist el Digiton Digimedia i les seves capes de consulta i mostrat d'imatges, presentem els requisits principals que el present projecte vol millorar:

Requisit 1:

L'eina ha de permetre la recuperació d'imatges provinents de fotogrames d'un *asset* complet a partir d'una idea de l'usuari.

Requisit 2:

L'eina ha de poder treballar sobre la base de dades d'un conjunt de test reduït de dades exportades del Digiton Digimedia i ha de ser una mostra representativa de diferents temàtiques com notícies, esports, canal parlament i programes.

Requisit 3:

L'eina ha de ser com a mínim més ràpida en la consulta mitjançant paraules text que qualsevol altra eina que actualment disposin els usuaris finals de la Corporació per a la recuperació d'imatges. Per això ha de poder suggerir paraules claus alternatives

emprant o bé el conjunt controlat de paraules del Thesaure de la Corporació (ontologia de paraules que porta 21 anys sent actualitzada per documentalistes de TV3) o el servei recomanador de paraules de Yahoo! Research, empresa amb qui la Corporació ha participat en la recerca del projecte subvencionat europeu Semedia⁵³.

Requisit 4:

L'eina ha de poder ser accessible des de qualsevol ordinador dins o fora de la Corporació indiferentment del sistema operatiu o el navegador que s'utilitzi.

Requisit 5:

L'eina ha de poder associar les paraules text amb els fotogrames **independentment** de si aquestes imatges tenen associades algun tipus de descriptors (baix, mitjà o alt nivell) com si disposen de cap descriptor associat.

Requisit 6:

L'eina ha de guanyar en facilitat d'ús i ha de ser intuïtiva en la seva representació dels fotogrames permetent mostrar fotogrames associats a un sol *asset*, a més d'un *asset* o intercalats entre *assets* diferents.

⁵³ www.semedia.org

7 ARQUITECTURA DEL SISTEMA

7.1 INTRODUCCIÓ

En el passat capítol de l'Estat de l'Art s'han introduït dos conceptes tant diferents com les consultes d'arxius Media emprant descriptors textuals i/o d'imatge i les aplicacions d'Internet enriquides basades en el concepte de programació als núvols. En aquest capítol es presenta l'eina desenvolupada en el marc d'aquest projecte així com les solucions que aquest projecte aporta sobre cadascun dels requisits presentats anteriorment.

Per tal d'aconseguir resoldre els requisits prèviament esmentats, cal definir els objectes de valor (*Value Objects*) bàsics com a dades de partida del programa. El principals objectes de valor són l'*asset* i l'*estrat*. Es pot trobar una descripció visual dels mateixos a la Figura 32.

Un *asset* és un element digital transversal en diferents programes de la Corporació. Representa un conjunt d'elements Media i un conjunt de descriptors que els descriuen. Els *assets* poden ser utilitzats en diferents punts del treball d'un usuari de la CCMA, ja sigui durant la ingesta (traducció d'un material analògic a digital) o com a escissió d'un editat pare. Un *asset* està determinat unívocament per un identificador anomenat *assetId*.

Els *estrats* són metadades textuals que descriuen un segment temporal de l'*asset* amb un codi temporal d'inici i un codi temporal final. Si el codi temporal d'inici i el final coincideixen, l'*estrat* és una descripció d'un sol fotograma.

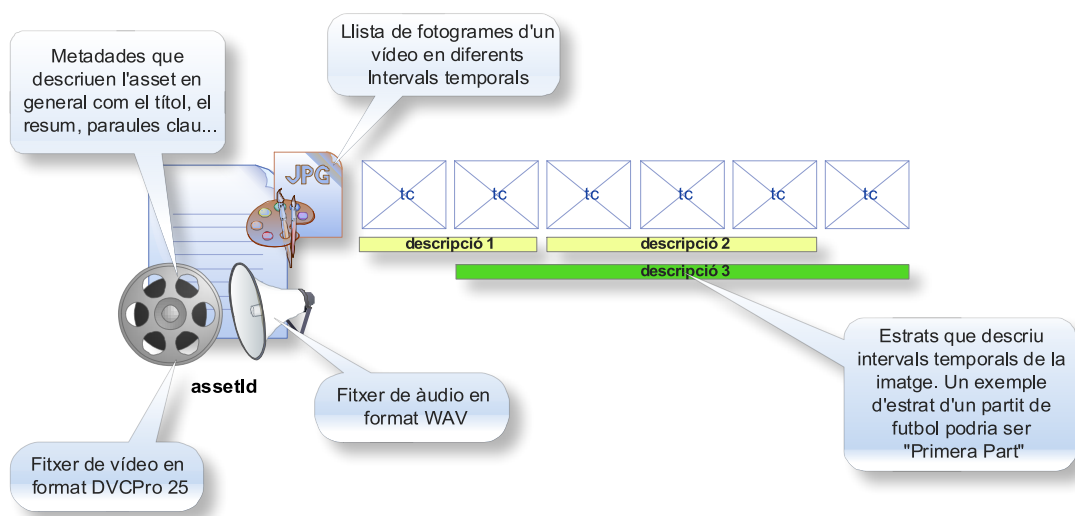


FIGURA 31 DESCRIPCIÓ D'UN ASSET I UN ESTRAT

Així doncs, l'aplicació web rica ha de poder importar i treballar amb els *assets* ja generats. De fet, s'entén que les respostes del sistema al requisit 1 es generen a partir d'aquest valor identificador (*assetId*) i apuntant a un TC (*Time Code*) d'una de les seves imatges.

7.2 ENTORN DE PROGRAMACIÓ EMPRAT

Abans de presentar l'estructura de l'aplicació principal del projecte es descriuen, a mode de resum, els entorns de programació, els connectors i els llenguatges amb les quals s'ha desenvolupat l'eina.

En primer lloc, un dels dos grans entorns de programació emprats és el conegut IDE **Eclipse** (versió Ganymede) juntament amb el connector **JavaEE** (Enterprise Edition) per incloure a la programació dels recursos servidor. JavaEE es diferencia de JavaSE (Standard Edition) per incloure el connector WTP (Web Tools Platform), funcionalitat per servir aplicacions Java sobre servidors. El servidor que s'ha utilitzat és el **Apache Tomcat** ⁵⁴ en la seva versió 6.

L'altre gran entorn de programació és el relacionat amb el client. Aquest és **Flex** en la seva tercera versió. Flex permet compilar dos tipus de llenguatges diferents, l'**Action Script 3** i l'**MXML**.

- ✓ L'MXML és un derivat de l'XML que permet definir fàcilment elements visuals de forma programàtica.
- ✓ L'AS3 és un derivat de Java Script per a ser interpretada pel connector Flash d'Adobe i permet crear les variables i donar lògica als elements visuals definits en l'MXML.

Flex 3 consta d'un paquet SDK que ha anat evolucionant en forma d'actualitzacions, des de la 3.1 fins la 3.4. Adobe ha preparat un connector per l'entorn de programació Eclipse que permet tenir en una sola IDE els codis del client i el servidor.

La representació dels objectes de valor es fa mitjançant objectes Java (si es treballa l'objecte al servidor Java) o objectes Action Script 3 (si es treballa l'objecte al client Flex) que conté descripcions incloses a la Taula 1.

⁵⁴ tomcat.apache.org

Nom propietat	Tipus propietat Java	Tipus propietat AS3	Descripció
assetId	Int	Number	Identificador unívoc de l' <i>asset</i>
Títol	String	String	Títol principal de l'aset
Resum	String	String	Resum principal de l' <i>asset</i>
hasStrats	Boolean	Boolean	Cert sí té strats o false si no en té
hasRelevantTitol	Boolean	Boolean	Cert si el títol conté la paraula amb què s'ha iniciat la cerca
hasRelevantResum	Boolean	Boolean	Cert si el resum conté la paraula amb què s'ha iniciat la cerca
strats	ListArray	ArrayCollection	Conjunt d'estrats per a un <i>asset</i>

TAULA 1 DESCRIPCIÓ, TIPUS I EQUIVALÈNCIA D'OBJECTES TIPUS ASSET

Les metadades dels *assets* es troben emmagatzemades en una base de dades Informix de l'empresa IBM en la taula **dtvc_arxiugeneral**.

Els *estrats* són metadades generades manualment o automàticament. Un *asset* pot tenir més d'un estrat. Els estrats estan identificats per un codi de temps (*timecode*) d'entrada i un timecode de sortida. El seu modelatge és mostra en la Taula 2

Nom propietat	Tipus propietat Java	Tipus propietat AS3	Descripció
Description	String	String	Contingut textual de l'estrat
Tcin	String	String	Cadena d'entrada en format hh:mm:ss:ff
Tcout	String	String	Cadena de sortida en format hh:mm:ss:ff
isRelevant	Boolean	Boolean	Cert sí conté la paraula amb què s'ha iniciat el filtre d'estrat.

TAULA 2 DESCRIPCIÓ, TIPUS I EQUIVALÈNCIA D'OBJECTES TIPUS ESTRAT

Les metadades dels estrats es troben emmagatzemades en una base de dades Informix en la taula **dtvc_arxiugeneral_description**.

7.3 DESCRIPCIÓ DELS RECURSOS EN EL PLA SERVIDOR

En aquest apartat es mostra l'estructura del sistema general per tal de donar suport a l'aplicació desenvolupada. En els diferents apartats, es justifica quins son els motius per a escollir les tecnologies a les quals s'ha optat.

Aquest capítol també presenta les complicacions i a la vegada les solucions que comporta un sistema enriquit per múltiples fonts de dades diferents i de naturalesa diferent. Es defineixen, doncs, aquestes fonts i s'explica com s'hi accedeix, com es tracten i quina funció tenen en el programa.

7.3.1 GRÀFIC GENERAL

A la Figura 32 es dóna una visió global dels recursos del sistema que impliquen el programa en la seva part servidora principalment. Per simplificar la descripció dels diferents mòduls i quina funció fan en el total de l'aplicació es nombren els diferents punts que recurrentment s'aniran tractant durant el present capítol. Es presenta la capa servidora en primer lloc i el conjunt de dades de múltiples fonts que hauran de ser lliurades als clients.

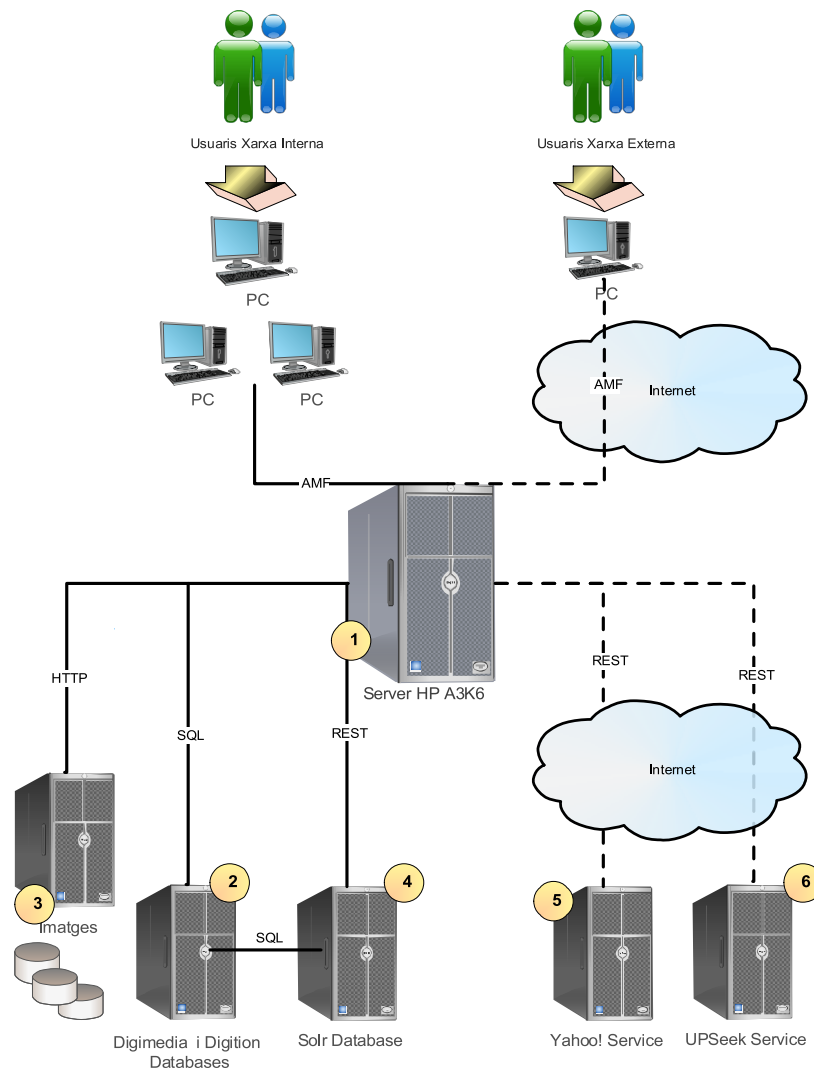


FIGURA 32 DIAGRAMA DE RECURSOS DEL SISTEMA

Es pot observar que el servidor principal és l'A3k6. Aquest servidor connecta amb els diferents recursos tant de la xarxa interna de la CCMA com els recursos externs. Aquests recursos externs són, per una banda, el servei web creat per Yahoo! de recuperació de paraules similars, i el servei web de la UPC de recuperació d'imatges similars. El servei de Yahoo suggereix paraules similars a la consulta entrada mentre que el servei de la UPC permet fer un reordenació d'imatges a partir de les seves característiques visuals.

També es pot observar com els usuaris que volen accedir als diferents serveis demanen accés a l'A3k6 per a utilitzar-los. Aquests usuaris poden estar dins o fora de la xarxa interna de la CCMA.

7.3.2 A3K6 (1), EL SERVIDOR NUCLI

El servidor **A3k6** (1) es troba en les dependències de l'edifici Imagina de la Corporació. Les seves principals funcions són les de gestionar totes les peticions que l'usuari fa per rebre les dades necessàries i fa d'hoste de l'aplicació final. El servidor es troba en una xarxa privada de la corporació (línia continua).

Aquest servidor és governat per una distribució Linux Ubuntu 9.04 i s'encarrega de cercar sobre la base de dades del Digition mitjançant crides **SQL**, crides **AMF** o bé crides **REST**, depenent del servei amb qui interaccioni. Per poder executar aquestes crides s'ha decidit emprar un servidor Apache Tomcat sobre el port 8080. És a dir, qualsevol dada de l'A3k6 cap a qualsevol client haurà de respondre a crides sobre aquest port.

La **Base de Dades Informix** (2) conté tota la informació dels *assets* del Digition Digimedia, així com les seves metadades. El servidor A3k6 (1) també s'encarregarà de parlar amb els serveis externs i de generar els fotogrames i guardar-los per a la seva futura recuperació.

7.3.3 APACHE SOLR (4), RECUPERACIÓ OPTIMITZADA DE METADADES

Inicialment, durant l'evolució del projecte, l'accés de l'A3k6 sobre les dades dels *assets* a cercar en la Base de Dades Informix(2) es feia directament mitjançant un controlador **jdbc** específic per a aquesta taula. Detalls de la instal·lació del controlador per Java i les seves particularitats es troben a la web d'IBM⁵⁵.

Ara bé, les cerques sobre els camps **Títol**, **Resum** o **Estrata** es van demostrar inviables si s'utilitzava una sentència SQL del tipus *Select* emprant la comanda **Like** i una paraula clau. De fet s'han provat mètodes de sentències que agilitzessin les crides sobre els camps esmentats com l'ús de caràcters comodí (wildcards) '%', '&' o '*'. Es va comprovar però que la falta d'un índex (reestructuració ordenada de la informació textual) de les bases de dades feia massa lent portar aquest tipus de sentències sobre conjunts grans d'informació, com els **454626 assets** amb els que es pretén treballar.

⁵⁵ www-01.ibm.com/software/data/informix/tools/jdbc/

En la cerca d'una solució per a aquest problema s'ha optat per emprar l'eina **Apache Lucene**⁵⁶ i el seu encapsulat en un mòdul Java viable per a ser funcional sobre un servidor Apache. L'aplicació completa s'anomena **Solr**. El seu funcionament dota d'un mecanisme de cerca ràpida sobre el conjunt de metadades emprant un sistema modular, actualitzable i que permeti fer cerques estandarditzades. De fet, la necessitat de crear una aplicació per a diferents sistemes operatius es soluciona gràcies a l'ús de la tecnologia Java, fàcilment portable a qualsevol programa existent i les cerques estandarditzades gràcies al protocol REST. És a dir, amb un senzill navegador podem preguntar i llegir els resultats que ens són enviats o bé en format **XML** o bé en format **JSON**.

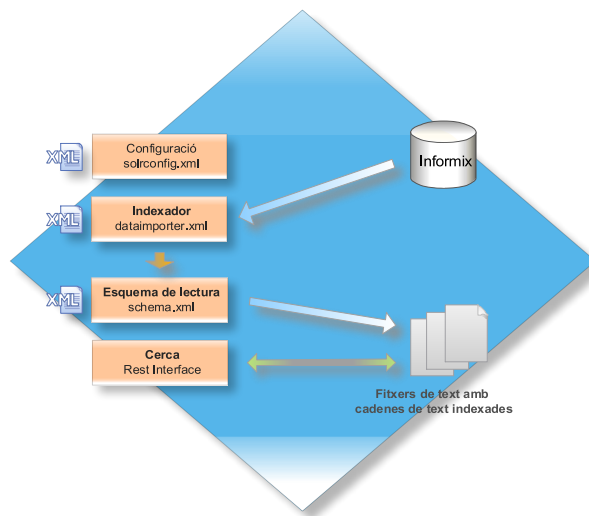


FIGURA 33 CONFIGURACIÓ DE L'APACHE SOLR

Així doncs, es va optar per substituir el controlador (*driver*) JDBC que s'emprava directament des del servidor al nou *servlet* (programa java dins d'un servidor) de Solr. Això permet que aquest accedeixi a les taules Informix i les indexi de forma eficaç i intel·ligent per a una posterior cerca ràpida. La Figura 33 mostra com el sistema està totalment configurat mitjançant arxius XML. L'indexador controla les sentències SQL necessàries per a cridar a la base de dades, un cop importades, l'esquema dona forma i prioritza els camps que es volen indexar.

En la Figura 34 es mostra un exemple de cerca sobre Solr.

⁵⁶ lucene.apache.org/java/docs/

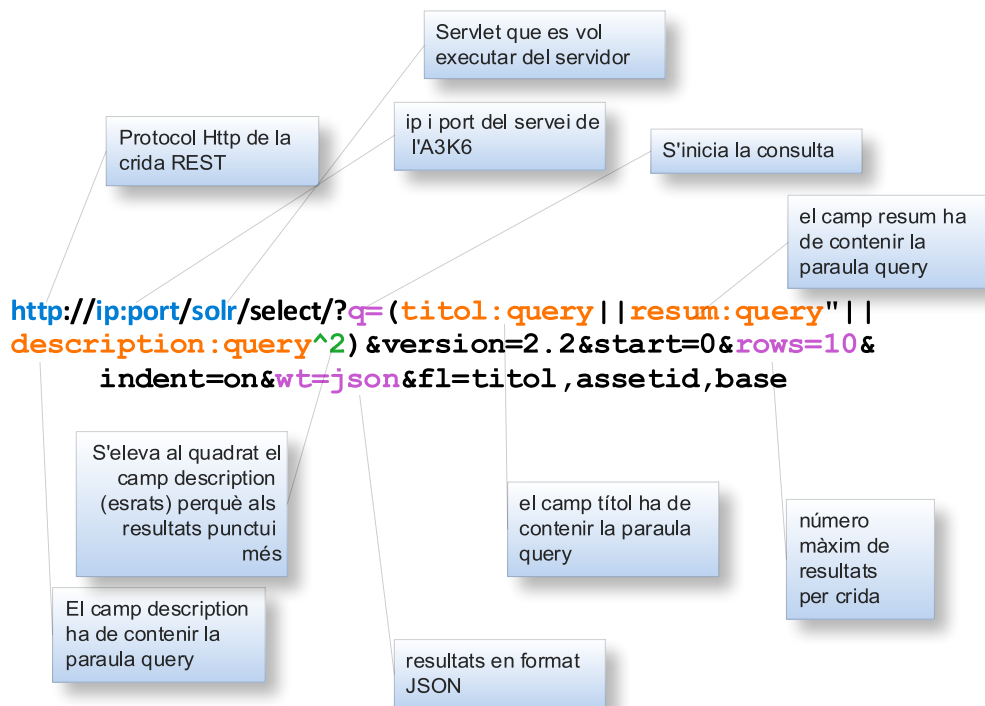


FIGURA 34 EXEMPLE DE CRIDA REST A SOLR

La cerca està codificada completament dins la crida HTTP. Primer s'indica la direcció del servidor Solr i tot seguit s'especifiquen els paràmetres que intervenen en la crida. En l'exemple s'està fent una consulta amb la paraula "query" com a exemple.

7.3.4 REPOSITORI D'IMATGES (3), L'EXTRACCIÓ D'IMATGES MITJANÇANT L'FFMPEG

Un *asset* conté un conjunt d'imatges clau que el representen. Aquests fotogrames es generen en uns temps concrets (*time codes*) durant la ingesta del vídeo dins del Digiton Digimedia. La llista d'interval des d'on s'agafa un fotograma es guarda directament en una taula dins de Digiton Digimedia i s'hi pot accedir fàcilment emprant la següent funció SQL donant un *assetId* o una llista d'*assetIds*.

EXECUTE FUNCTION OBTENIR_KF_LLISTAASSETS_STRING(LIST{"+ASSETID+"}).

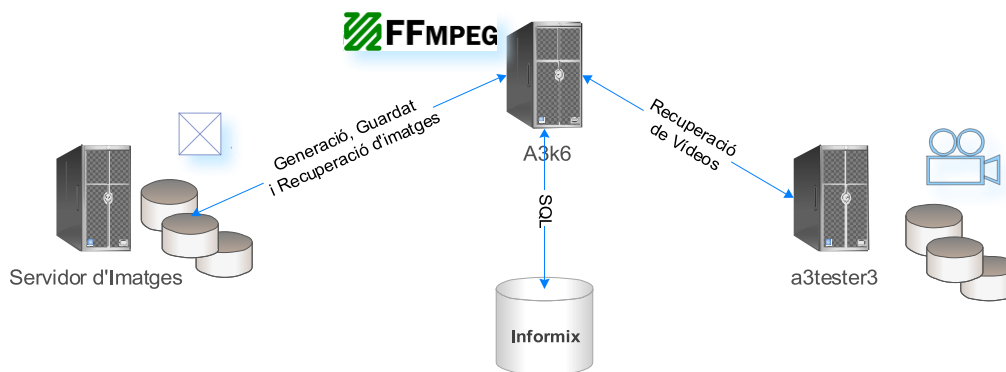
Aquesta consulta retorna una llista de temps en format **hh:mm:ss:ff** on **ff** és un dels 25 fotogrames per segon.

El procés d'ingesta no només identifica els *time codes* més representatius de la imatge sinó que també genera imatges en miniatura (*thumbnails*) que es guarden en una base de dades específica governada per un servidor Web DataBlade d'Informix (servidor privatiu creat sobre un servidor Apache).

Tanmateix, aquest servidor no és accessible per aquest projecte i no es pot redirreccionar pel port de sortida 8080. Per aquest motiu, es fa necessari desenvolupar una eina d'extracció de fotogrames directament dels fitxers de vídeo del subconjunt Digimedia. Per fer-ho cal tenir en compte que es tracta de fitxers codificats en format **DVCPRO25** de l'empresa Panasonic dins d'un contenidor AVI. Els fitxers de vídeo en DVCPRO25 són guardats en cintes d'ampli ús en les càmeres professionals del sector audiovisual.

L'extracció d'imatges es realitza amb el programa lliure **FFMPEG**⁵⁷ que genera els fotogrames segons els codis de temps recuperats de la base de dades. S'ha escrit una aplicació amb Java que realitza dues funcions diferents (com es visualitza en la Figura 35) :

- Crida a la base de dades per recuperar una llista de fotogrames i el directori amb el fitxer AVI donat un *assetId*.
- Entrega la llista i la direcció del vídeo al programa FFMPEG creant dos fills (threads) diferents. Un per a l'execució del programa i l'altre per a llegir-ne la seva sortida estàndard.



Vídeos		TimeCodes	
assetId	Url	assetId	TimeCode
1004	\\a3tester3\files\videos\1004\1004_0473729.avi	1004	00:00:00:00
		1004	00:00:05:13
		1004	00:00:32:03

FIGURA 35 EXTRACTOR DE FOTOGAMES

La comanda de la Figura 36 (només vàlida per a màquines en entorn GNU/Linux amb el paquet FFMPEG instal·lat prèviament) mostra la generació de *keyframes* donat un tc "timecode" (cal passar de hh:mm:ss:ff a milisegons) i una direcció UNC o SAMBA a un vídeo "videoname".

⁵⁷ ffmpeg.org/

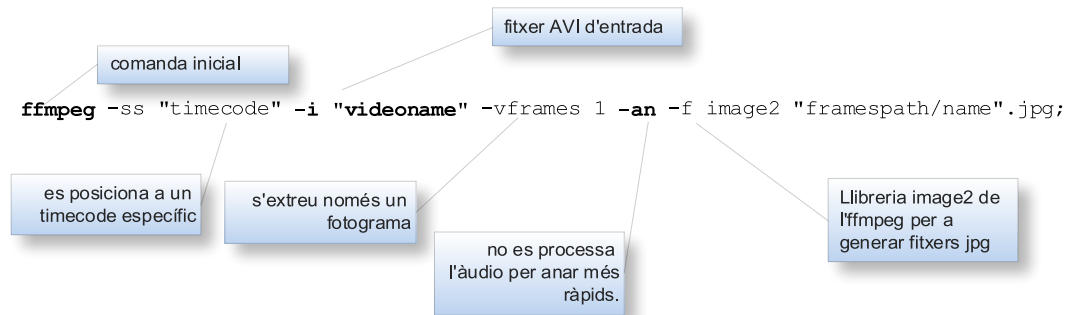


FIGURA 36 COMANDA FFMPEG UTILITZADA

Un cop generats els fotogrames, cal crear un *servlet* específic dins del servidor general A3k6 variant el context general del servidor en el seu arxiu de configuració **server.xml** i utilitzant el recurs ja implementat **Directory** per a servir fitxers directament d'una carpeta. Gràcies a aquest procediment, es pot visualitzar un fotograma directament amb una consulta a l'A3k6, per exemple:

HTTP://IP_A3K6:PORT/CONTEXT/ASSET/FOTOGRAMA.JPG

7.3.5 SERVEI WEB DE YAHOO! (3)

El servei Web de Yahoo! és una eina específica creada en el marc del projecte europeu Semedia⁵⁸ i optimitzada específicament per al repositori de metadades de la Corporació. Aquesta eina, després de fer un estudi estadístic del conjunt de metadades de l'arxiu, permet recuperar paraules recomanades i similars a d'altres a una donada.

A continuació es presenta un exemple pràctic: donada una crida REST similar a la que es pogués fer amb Solr però amb un sol paràmetre d'entrada (la paraula inicial), la crida respon amb un XML que conté paraules en un context similar. El XML de retorn per a la paraula "parlament" és el que mostra la Figura 37.

En cas de no trobar paraules recomanades, el servei genera una llista d'alternatives. Aquesta funcionalitat permet adreçar-se a casos com l'autocompletació de paraules. A la Figura 38 es mostra el XML de retorn a la paraula "zapatero".

⁵⁸ www.semedia.org


```

-<tags>
-<query taglist="parlament">
  <qtag tag="parlament" raw="parlament"/>
  </query>
  <stag tag="ple del parlament" raw="PLE DEL PARLAMENT" score="1"/>
  <stag tag="pais basc" raw="PAIS BASC" score="0.8"/>
  <stag tag="vitoria" raw="VITORIA" score="0.666666666666667"/>
  <stag tag="votacio" raw="VOTACIO" score="0.57142857142857"/>
  <stag tag="gb" raw="GB" score="0.5"/>
  <stag tag="lleis" raw="LLEIS" score="0.444444444444444"/>
  <stag tag="eua" raw="EUA" score="0.4"/>
  <stag tag="londres" raw="LONDRES" score="0.36363636363636"/>
  <stag tag="iraq" raw="IRAQ" score="0.333333333333333"/>
  <stag tag="manifestacio" raw="MANIFESTACIO" score="0.30769230769231"/>
  <stag tag="govern" raw="GOVERN" score="0.28571428571429"/>
  <stag tag="reunio" raw="REUNIO" score="0.266666666666667"/>
  <stag tag="ibarretxe, juan jose" raw="IBARRETXE, JUAN JOSE" score="0.25"/>
  <stag tag="terrorisme" raw="TERRORISME" score="0.23529411764706"/>
  <stag tag="diputats" raw="DIPUTATS" score="0.222222222222222"/>
  <stag tag="acte de protesta" raw="ACTE DE PROTESTA" score="0.21052631578947"/>
  <stag tag="atutxa, juan m." raw="ATUTXA, JUAN M." score="0.2"/>
  <stag tag="discurs politic" raw="DISCURS POLITIC" score="0.19047619047619"/>
  <stag tag="blair, tony" raw="BLAIR, TONY" score="0.181818181818181"/>
  <stag tag="pais valencia" raw="PAIS VALENCIA" score="0.17391304347826"/>
</tags>
- <!--
  s2.semedia.ukl.yahoo.com uncompressed/chunked Wed Oct 14 14:25:00 GMT 2009
-->

```

FIGURA 37 XML DE RESPOSTA AL SERVEI DE YAHOO!

```

-<tags>
-<query taglist="zapatero">
  <qtag tag="zapatero" raw="zapatero"/>
  </query>
-<alternatives>
  <atag tag="rodriguez zapatero, jose luis" raw="RODRIGUEZ ZAPATERO, JOSE LUIS"/>
  <atag tag=" rodriguez zapatero, jose luis" raw=" ROGRIGUEZ ZAPATERO, JOSE LUIS"/>
  <atag tag="zapatero gomez, virgilio" raw="ZAPATERO GOMEZ, VIRGILIO"/>
  <atag tag="rodriguez zapatero, jose luis pp" raw="RODRIGUEZ ZAPATERO, JOSE LUIS PP"/>
  <atag tag="rodriguez zapatero, jose luis" raw="RODRIGUEZ ZAPATERO, JOSE LUIS"/>
</alternatives>
</tags>
- <!--
  s2.semedia.ukl.yahoo.com uncompressed/chunked Wed Oct 14 14:07:05 GMT 2009
-->

```

FIGURA 38 EXEMPLE D'ALTERNATIVES

7.3.6 SERVEI WEB DE L'UPSEEK

Aquest servei té com a finalitat permetre les cerques per exemple emprant un motor de cerques de la UPC. Aquest motor permet recuperar fotogrames a partir d'una mesura de similitud entre descriptors visuals de les imatges.

El servei web de l'Upseek també és un servei REST similar al servei de Yahoo! En aquest cas s'han determinat diferents mètodes segons la informació que es vulgui recuperar.

La relació de mètodes és la mostrada en la Taula 3 :

Mètode	Paràmetres	Resposta	Descripció
http://ip:port/upseek-server/mediaassetexists?	corpus=ccma&assetid=1006&format=json	Boolean	Retorna cert si l'asset està processat pel servidor de la UPC.
http://ip:port/upseek-server/querybyexample?	corpus=ccma&asset=1003&keyframe=00_00_00_07&format=json	Array String	Retorna una relació de fotogrames semblants al fotograma d'exemple.
http://ip:port/upseek-server/mediaassetslist?	corpus=ccma&format=json	Array String	Retorna la llista completa d'assets processats pel servidor de la UPC.
http://ip:port/upseek-server/mediahowmanyassets?	corpus=ccma&format=json	Integer	Retorna el número d'assets processats.

TAULA 3 MÈTODES REST DE L'UPSEEK

El mètode clau és el **QueryByExample?** Un exemple de fitxer XML de resposta és el de la Figura 390, on es pot comprovar que la resposta és un array de fotogrames amb una puntuació (score) determinada pel servidor Upseek.

```

- <result>
  - <array>
    <corpus>ccma</corpus>
    <asset>1003</asset>
    <keyframe>00_00_00_07</keyframe>
    <score>0.999999</score>
  </array>
  - <array>
    <corpus>ccma</corpus>
    <asset>1003</asset>
    <keyframe>00_00_01_03</keyframe>
    <score>0.904688</score>
  </array>
  - <array>
    <corpus>ccma</corpus>
    <asset>1003</asset>
    <keyframe>00_00_01_11</keyframe>
    <score>0.9</score>
  </array>

```

FIGURA 39 RESPOSTA AL MÈTODE QUERYBYEXAMPLE

7.3.7 PROBLEMES DE SEGURETAT DEL CONNECTOR FLASH PER A SERVEIS EXTERNES

En un primer disseny de l'aplicació els clients directament implementaven les crides REST per ambdós serveis externs, Upseek i Yahoo!. Tanmateix, per motius de seguretat del connector Flash⁵⁹, aquesta solució no és viable. L'explicació d'aquest fet recau en les fortes mesures que seguretat que implementa el connector Flash (Flash Security

⁵⁹ kb2.adobe.com/cps/142/tn_14213.html

Sandbox). Aquestes mesures fan que només es permeti descarregar informació des del servidor d'on s'ha baixat l'aplicació Flash. S'eviten així possibles suplantacions d'identitat dels servidors de dades o permetent l'accés de dominis no segurs. En aquesta primera figura es pot observar que si un client accedeix a una aplicació Flash d'un servidor es crea un camí únic segur entre les dos màquines (Figura 42).



FIGURA 40 ACCEDINT A L'APLICACIÓ, CREAT UN CAMÍ SEGUR

És a dir, si es vol utilitzar l'aplicació flash App.swf que es troba al servidor A3k6 accessible mitjançant la direcció http://a3k6_ip:a3k6_port/app.swf i després es pretén passar informació de l'a3k6 cap al client que està executant l'aplicació, **no** hi haurà problemes de seguretat (veure Figura 41). Però si s'intenta accedir directament a les dades de la font d'informació de Yahoo! sense passar pel servidor a3k6 ens apareixeran avisos o fins i tot errors de compilació avisant de forats de seguretat en l'aplicació.

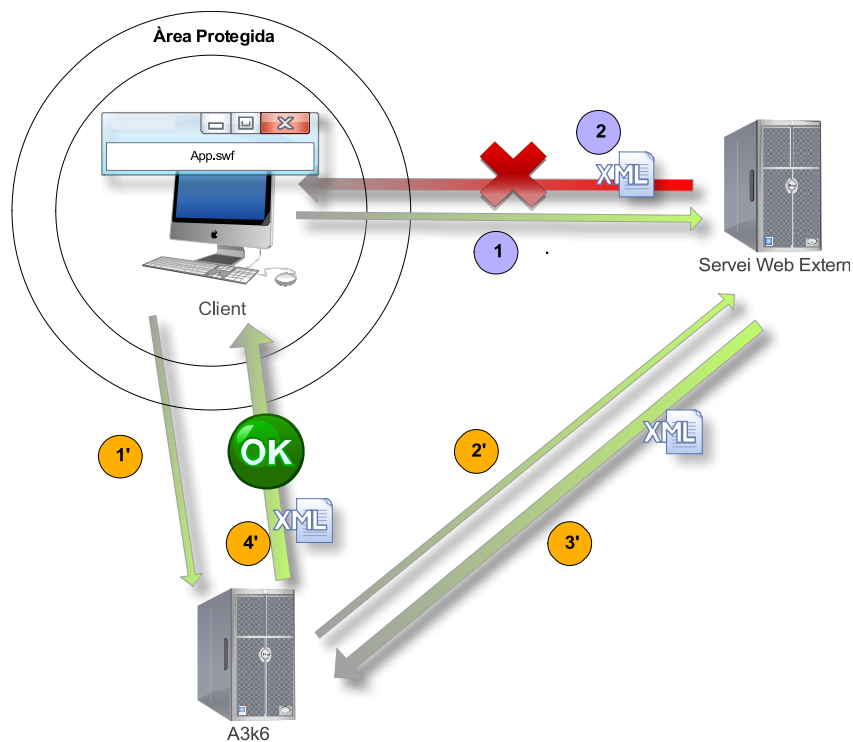


FIGURA 41 RECUPERACIÓ CORRECTA D'INFORMACIÓ EXTERNA

La solució de disseny adoptada finalment fa que les crides als serveis web externs siguin generades directament pel servidor A3k6 (1), i que el fitxer XML de retorn sigui enviat al client pel mateix servidor.

7.3.8 CONNEXIÓ AMB EL CLIENT, IMPLEMENTACIÓ DEL PROTOCOL AMF

L'últim canal de comunicació que resta per definir és el canal que comunica el client amb el servidor. Aquest canal és el que suporta més informació i ha de permetre múltiples crides asíncrones. El protocol escollit és AMF i té dues implementacions, una comercial i l'altre oberta anomenada **BlazeDS**.



BlazeDS dóna tota una infraestructura comunicativa sobre el protocol HTTP. De fet, dóna la opció d'implementar tres tipus de serveis diferents: el servei Remot, el Servei Web i el Servei de Missatgeria. En aquesta aplicació s'emprarà el primer, el servei Remot.

BlazeDS serialitza els objectes Java i Flex d'un cantó a l'altre del pont remot, guardant les seves propietats. La serialització duu inclosa una compressió intel·ligent i redundat que la fa forta a caigudes de xarxa, a més de fer possible la encriptació de les dades.

A la Figura 42, extreta del manual oficial del desenvolupador⁶⁰, es poden veure els tres tipus diferents de serveis que ofereix BlazeDS i s'ha remarcat el servei remot AMF que s'empra en el present projecte.

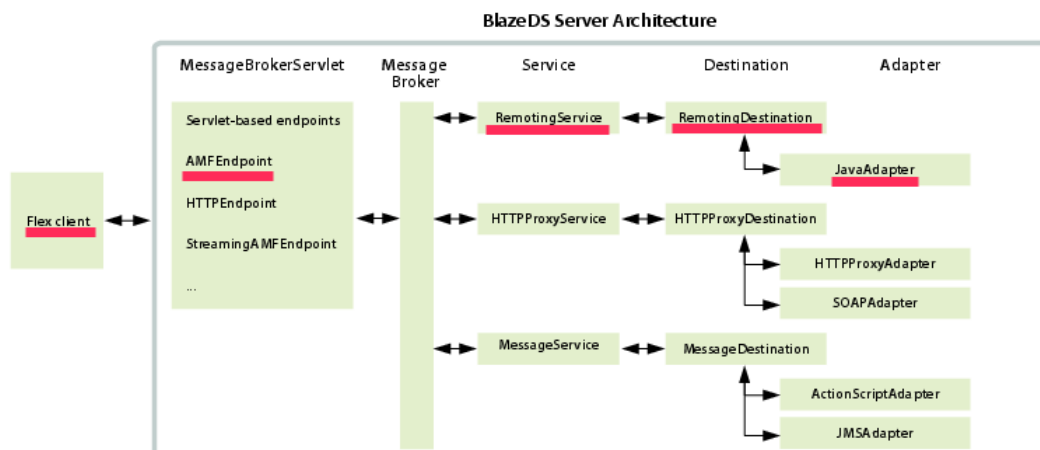


FIGURA 42 ARQUITECTURA DE LES COMUNICACIONS AMB BLAZEDS

Cal doncs definir a nivell de servidor el nostre AMF Endpoint i un canal dins de l'escoltador (*Broker*) que connectarà amb les nostres classes Java.

⁶⁰ livedocs.adobe.com/blazeds/1/blazeds_devguide/

Una crida que vingui de Flex i sigui escoltada pel Broker instanciarà una classe Java, executarà un mètode de la classe i finalment en retornarà el seu possible resultat. Tot això sent transparent per l'usuari un cop s'hagin definit els XML de configuració pertinents.

En l'arxiu **proxy-config.xml** (Figura 43) es defineix el canal AMF que s'anomenarà my-amf:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <service id="proxy-service" class="flex.messaging.services.HTTPProxyService">
- <properties>
- <connection-manager>
  <max-total-connections>100</max-total-connections>
  <default-max-connections-per-host>2</default-max-connections-per-host>
</connection-manager>
  <allow-lax-ssl>true</allow-lax-ssl>
</properties>
- <adapters>
  <adapter-definition id="http-proxy" class="flex.messaging.services.http.HTTPProxyAdapter" default="true" />
  <adapter-definition id="soap-proxy" class="flex.messaging.services.http.SOAPProxyAdapter" />
</adapters>
- <default-channels>
  <channel ref="my-amf" />
</default-channels>
  <destination id="DefaultHTTP" />
</service>
```

FIGURA 43 PROXY-CONFIG.XML

I dins del fitxer **remoting-config.xml** (Figura 44) definirem les relacions entre les crides AMF i les classes Java que han d'instanciar:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <service id="remoting-service" class="flex.messaging.services.RemotingService">
- <adapters>
  <adapter-definition id="java-object" class="flex.messaging.services.remoting.adapters.JavaAdapter" default="true" />
</adapters>
- <default-channels>
  <channel ref="my-amf" />
</default-channels>
- <destination id="thesaureservice" channels="my-amf">
- <properties>
  <source>com.ccma.thesaure.BDThesaure</source>
</properties>
</destination>
- <destination id="httpservice" channels="my-amf">
- <properties>
  <source>com.ccma.serveisweb.HttpRequest</source>
</properties>
</destination>
</service>
```

FIGURA 44 REMOTING-CONFIG.XML

Podem observar que tenim dues classes per instanciar: La primera és la **com.ccma.thesaure.BDThesaure** que s'encarrega de fer crides a la base de dades Informix per a recuperar les paraules de la CCMA que no han sigut indexades per Solr.

La segona és la **com.ccma.serveisweb.HttpRequest** que s'encarrega de processar una crida Rest ja sigui a l'aplicació Solr o als serveis web externs i en retorna els resultats pel canal AMF.

A nivell de client (que es tractarà en el següent capítol) el protocol AMF ve integrat en el SDK de FLEX, però a nivell de servidor cal copiar un seguit de fitxers JAR (classes Java compilades i agrupades) dins de la carpeta LIB de l'aplicació servidora.

Amb tot, tenim un servidor multifont preparat per escoltar crides AMF provinents de múltiples clients Flex que demanin informació per a la seva aplicació.

7.4 DESCRIPCIÓ DELS RECURSOS EN EL PLA CLIENT

7.4.1 INTRODUCCIÓ AL DIGIMATGE

Un cop definits els recursos en el pla servidor així com les diferents fonts de dades de què disposem es procedeix a explicar els recursos que s'han emprat per poder fer accessible a l'usuari aquesta informació.

En aquest capítol, s'anirà presentant la nova eina enriquida: el Digimatge (Figura 456). Centrant-nos en les eleccions que s'han pres tant en el seu disseny com les dificultats en la posada en pràctica amb un entorn específic per a la seva programació, Flex.

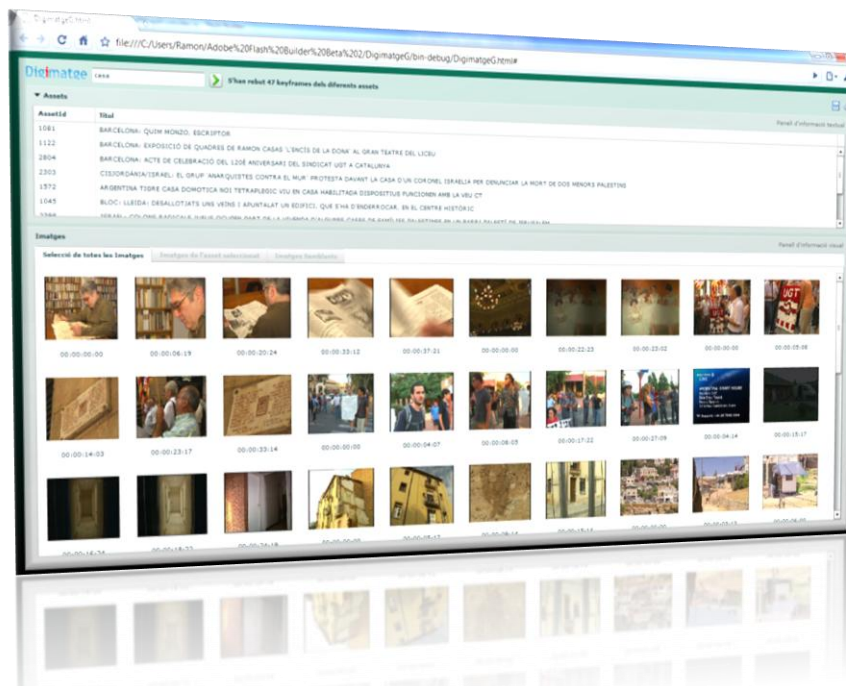


FIGURA 45 DIGIMATGE

7.4.2 MODEL DEL PROGRAMA CLIENT

El programa està dissenyat per objectes seguint una estructura MVC (Model Viewer Controller). Aquest model facilita la tasca del programador separant per capes els objectes segons la seva funcionalitat dins del programa. Aquestes funcionalitats poden ser:

- ✓ **Model**, s'encarrega de representar específicament les dades així com de mantenir la seva integritat al llarg del seu ús. El model és únic en el programa.

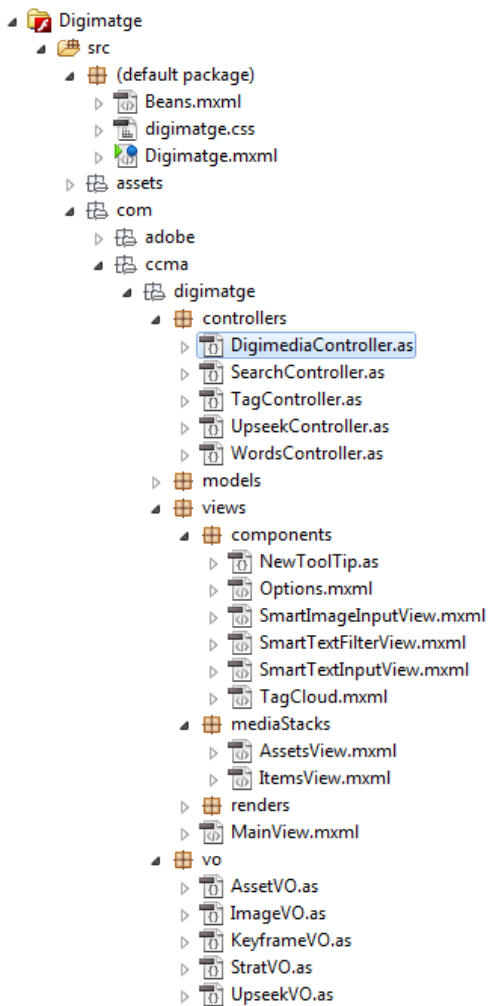


FIGURA 46 ESTRUCTURA DE FITXERS DEL PROGRAMA

✓ **Controladors**, en aquest programa, s’encarreguen de fer les connexions a l’exterior de l’aplicació, mantenint les polítiques de sincronització amb les dades i guardant-ne còpia de les rellevants dins del model.

✓ **Vistes**, es limiten a visualitzar les dades del model i a escoltar les accions de l’usuari, passant les rellevants als controladors pertinents.

Hi ha diferents llibreries per a la creació de programes seguint el Model MVC, com Adobe Caingorm, Mate o **Swiz**. Sent aquesta última la llibreria escollida per seguir una metodologia d’inversió de control (IoC) molt útil pel programador. La inversió de control permet “injectar” objectes des del Model fins a les Vistes o el Controlador de mòduls només en el moment de ser utilitzats, fent l’aplicació més escalable i més òptima. En les Figura 467 i 48 es resumeixen tots els mòduls que involucren el programa, especificant quins mòduls pertanyen al model (*models*), quins al controlador (*controllers*) i quins a les vistes (*views*).

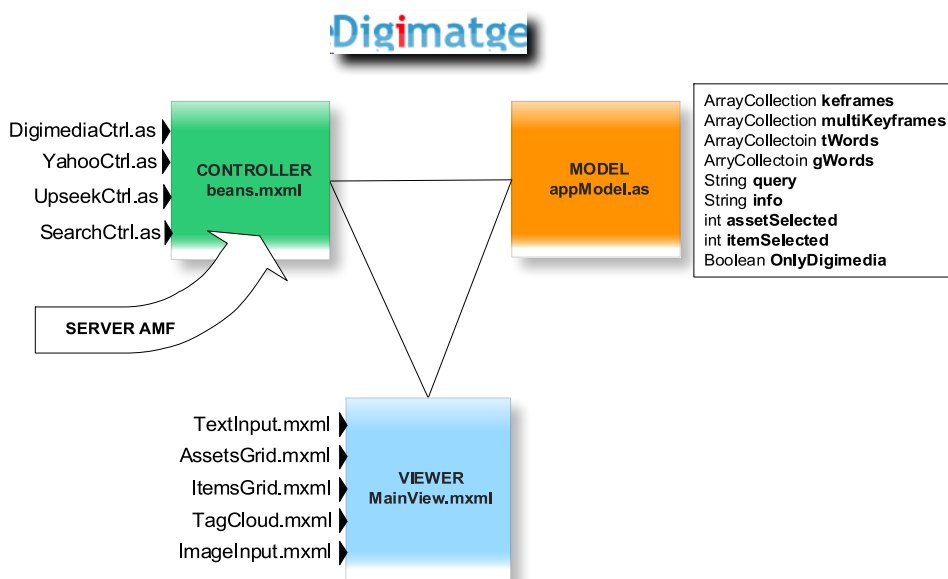


FIGURA 47 RELACIÓ ENTRE ELS MÒDULS DEL PROGRAMA

7.4.3 BEANS.MXML, CONFIGURACIÓ DE LES COMUNICACIONS I ELS CONTROLADORS

El fitxer **Beans.xml** conté tots els objectes remots que permeten gestionar les crides AMF que fan de nexa amb el servidor A3k6.

Primer cal definir un únic canal dinàmic que enviarà i escoltarà informació del Broker del servidor. La definició d'aquest canal es fa en llenguatge MXML.

```
<DynamicChannelSet id="myAmf">
  <serverName>{appModel.ip_CCMA}</serverName>
  <serverPort>{appModel.port_CCMA}</serverPort>
  <contextRoot>/DigimatgeServer</contextRoot>
</DynamicChannelSet>
```

Un cop definit el canal, creem objectes remots específics tot indicant quin canal dinàmic han d'emprar (*channelSet*), a quin destí remot s'han de connectar (*destination*), quin tipus de concurrència han d'utilitzar (*concurrency*), si el seu ús porta implicat un cursor en forma de rellotge (*showBusyCursor*). Opcionalment s'indiquen els mètodes viables de la classe remota que podran executar (*methods*).

```
<mx:RemoteObject id="thesaureService"
  destination="thesaureservice"
  showBusyCursor="true"
  concurrency="last"
  channelSet="{myAmf}">
  <mx:method name="getCandidates"/>
</mx:RemoteObject>
```

Com a punt interessant cal comentar els tres tipus de concurrència a què opten els Objectes Remots AMF. Poden ser *"last"*, *"single"* o *"multiple"*:

- ✓ **Last:** Es permeten múltiples crides al servidor, però qualsevol nova crida ignora els possibles resultats del servidor de les crides anteriors. Només els events de resultat o error de la última crida són llançats.
- ✓ **Single:** Només es permet llençar una crida a la cua de crides, qualsevol intent de llençar una crida subsegüent a la primera és contestat amb un error al client.
- ✓ **Multiple:** Es permeten múltiples crides a la cua, és el desenvolupador qui ha d'assegurar la consistència de les dades que es retornen.

Els controladors també queden definits en el arxiu Beans.xml, però es desenvolupen en fitxers AS independents per a cada fitxer.

Per exemple, el controlador del Thesaure ha d'escoltar la variable **query** que es troba dins del model. Cada cop que la variable es modifica (una sola lletra) s'enviarà una petició al servidor per a buscar paraules dins del **Thesaure** de la Corporació que comencin amb aquella lletra.

El controlador s'ha d'encarregar de rebre aquesta llista de paraules i guardar-la ordenadament a la variable del model anomenada **tWords**. Aquesta llista (veure Figura 48 Recomanació de paraules) servirà per a poblar un objecte de la interfície del tipus combobox.

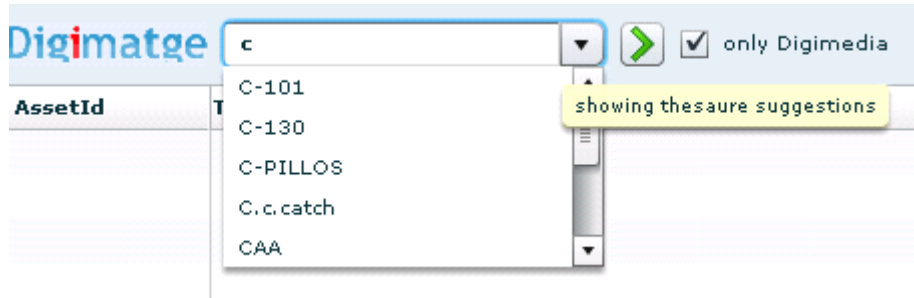


FIGURA 48 RECOMANACIÓ DE PARAULES

Cada nova lletra que l'usuari entri anul·larà la possible resposta del servidor per a la paraula **query** anterior. Això justifica que l'objecte remot tingui una concurrència del tipus **last**.

El codi més interessant del controlador és el següent:

```
executeServiceCall(thesaureService.getCandidates(query),
thesaureService_result, service_fault, null);
```

En aquesta sentència s'utilitza l'objecte remot **thesaureService** connectat al servidor. Es crida el mètode **getCandidates** tot passant la paraula **query** i s'indiquen els mètodes que haurà de cridar quan arribi asíncronament el possible resultat (mètode **thesaureService_result**) o el possible error (mètode **service_fault**).

7.4.4 APPMODEL, OBJECTES PERSISTENTS DURANT LA SESSIÓ DE L'APLICACIÓ

En la programa s'han introduït els objectes principals que són comuns a tota l'aplicació. Per aquest motiu el model es defineix per a ser accessible tant per les vistes com pels controladors. A continuació es defineixen els principals objectes i el tipus d'informació que emmagatzemen:

- ✓ **ArrayCollection keyframes:** Emmagatzema tants objectes del tipus **keyframeVO** com fotogrames tingui un sol *asset*. Els objectes de valor **keyframeVO** s'han definit en la introducció del present capítol.
- ✓ **ArrayCollection multipleKeyframes:** Emmagatzema tants arrays del tipus anterior com *assets* s'hagin trobat a la cerca per text. Això permet mantenir uns deu *assets* amb totes les seves urls dels fotogrames.
- ✓ **ArrayCollection upseekResults:** Emmagatzema tants objectes del tipus **keyframeVO** com resultats presenti la eina de consulta amb imatge.

- ✓ **ArrayCollection tWords:** Emmagatzema les paraules candidates a la **query** llançada.
- ✓ **String query** És la paraula que s'escriu en el text input de l'aplicació.
- ✓ **Int assetSelected** És la fila que l'usuari selecciona de la taula d' **assets**.
- ✓ **Int ItemSelected** És la fila que l'usuari selecciona de la taula de fotogrames
- ✓ **Boolean onlyDigimedia** Indica si la cerca es fa sobre el Digition Digimedia o sobre tot el Digition.

7.4.5 INTERFÍCIE DE L'APLICACIÓ

En aquest subapartat s'explica l'ús amb l'eina per tal de cercar *assets* o imatges indistintament utilitzant la informació textual (sobre els descriptors textuels que es disposen) o bé visual per a les consultes mitjançant la eina de cerca amb imatge de la UPC.

Inicialment l'usuari entra dins l'aplicació amb una idea aproximada del que vol buscar. Suposem l'exemple d'una imatge d'un àrbitre ensenyant una targeta groga.

L'usuari introduirà la paraula "targeta" al quadre inicial de text. Això iniciarà la cerca sobre les metadades (inicialment sobre els tres camps definits que són el títol, el resum o l'estrat, però que són configurables mitjançant el quadre d'opcions). Aquesta cerca respondrà amb un màxim de deu *assets* (valor que també es pot configurar al quadre d'opcions). També respondrà amb una previsualització dels deu primers fotogrames d'aquests *assets* tal i com es pot comprovar a la Figura 49.

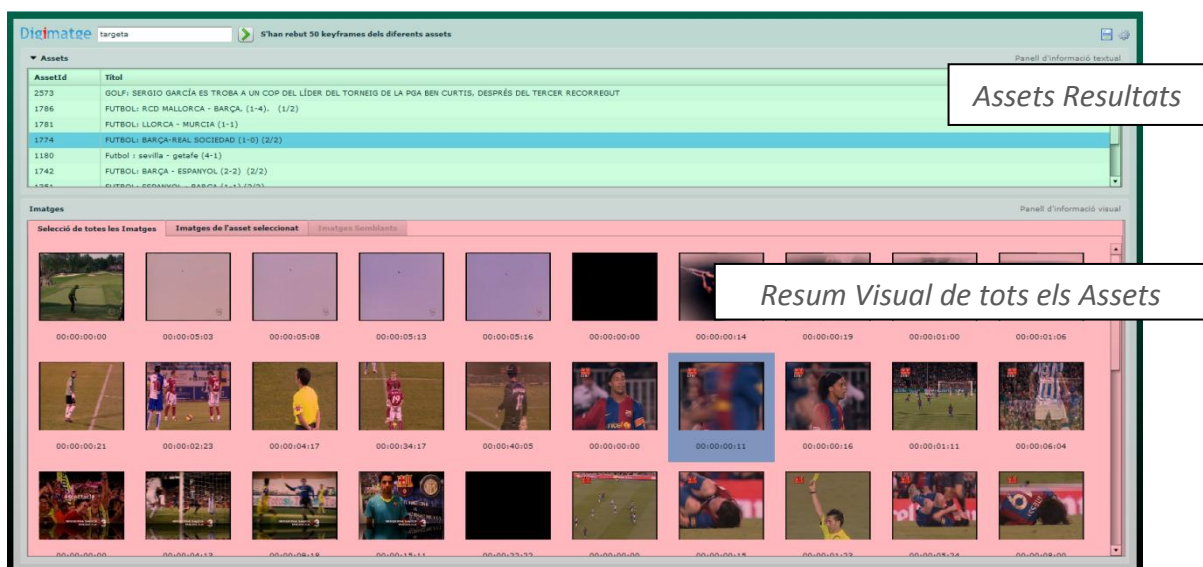


FIGURA 49 DIGIMATGE, PANTALLA PRINCIPAL

Fent doble clic sobre el algun dels *assets* resultat, es pot inspeccionar la informació textual annexada amb més detall. Aquest doble clic també mostra tots els fotogrames específics d'aquest *asset* (Figura 50).



FIGURA 50 DETALL DE L'ASSET

El quadre inferior (remarcat en blau) conté un segon quadre de text. Aquest quadre permet filtrar els fotogrames de l'*asset* segons una segona paraula trobada en els estrats. Per aconseguir unir estrats amb els fotogrames, prèviament s'annexen les imatges segons si el seu Time Code (tc) cau dins de l'interval **tcin-tcout** d'un estrat.

Si és així i conté la paraula del quadre de filtre, s'insereix una estrella a la cantonada superior dreta de la imatge. En la imatge superior es pot comprovar que la paraula "groga" mostra una imatge d'un àrbitre aixecant la mà ensenyant la cartolina.

Un altre exemple de l'ús d'aquest segon filtrat el trobem en *assets* que són gravacions plenàries del Parlament de Catalunya. La paraula inicial de consulta seria **Parlament** i la paraula de filtrat seria **Mas** (Figura 51) o **Carod**.

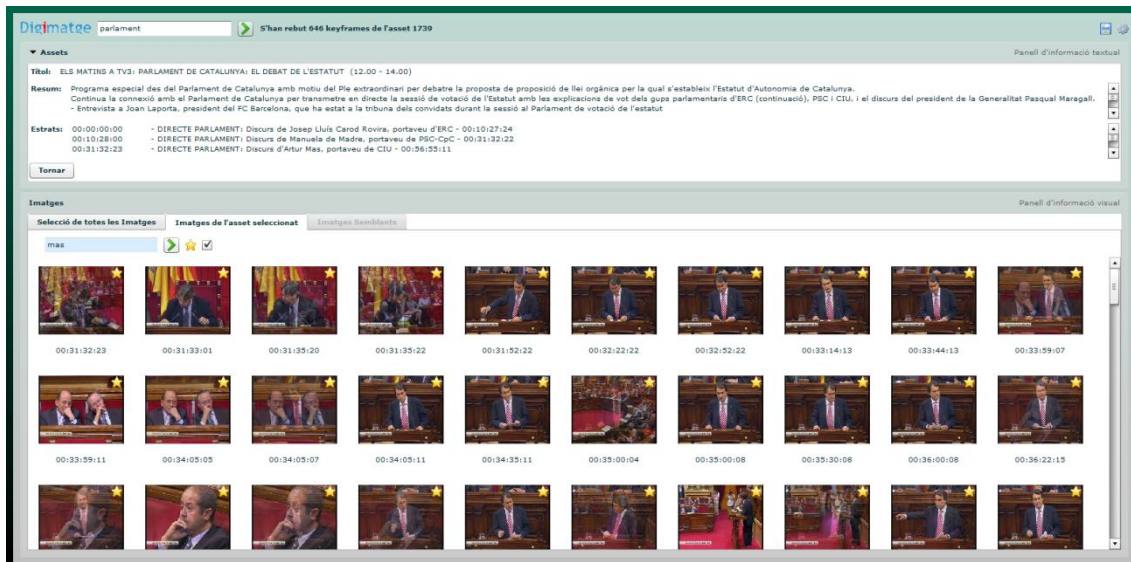


FIGURA 51 DETECCIÓ D'ARTUR MAS

Lògicament, aquesta aproximació de la cerca i filtrat sobre l'estrat està lligada al tipus de descriptors textuais que estan a l'abast en el moment de la realització de l'eina, és a dir, estrats manuals. L'aproximació també servirà per a futurs estrats generals automàtics i a nivell de fotograma.

La consulta visual es genera fent doble clic sobre qualsevol imatge. Això llença una petició sobre el servei de consulta visual de la UPC que retorna una llista de fotogrames ordenada de major a menor rellevància (semblança). Aquests fotogrames són de diferents *assets*. Així doncs, per no perdre el context d'un *asset* com a conjunt, per a cada nou *asset* diferent de la resposta de la UPC es llença una crida per tal de recuperar-ne el seu títol, el seu resum i els seus estrats.

La informació surt reflexada en el primer quadre d'*assets* permetent el doble clic per a recuperar només els fotogrames d'aquell *asset*.

8 TEST D'USUARI

8.1 INTRODUCCIÓ

En el aquest apartat es descriuen els resultats de diferents tests realitzats a usuaris professionals de la Corporació.

El perfil d'aquests usuaris és de dos tipus diferents:

- ✓ El primer perfil és el de **desenvolupador** de Digition, l'eina que el Digimedia tracta de millorar. Aquest perfil es considera avançat en l'ús d'eines informàtiques i per tant és una bona font per a la detecció de malfuncionaments de l'eina, millores en l'optimització i la detecció de punts forts i febles en comparació al Digition.
- ✓ El segon perfil és el de **documentalista** de Digition. Aquest perfil és el d'un usuari periodista dedicat exclusivament a la indexació i cerca d'assets dins del Digition. L'eina Digimatge ha estat concebuda principalment per solucionar les necessitats d'aquests usuaris.

Els tests d'usuari realitzats sobre l'equip desenvolupador han sigut més nombrosos i de més freqüència que els realitzats sobre els usuaris documentalistes. Aquest fet es deu a la ubicació i la proximitat disposada per tal de fer preguntes més o menys formals en les distintes etapes d'elaboració de l'eina. Per altra banda, s'han realitzat dos de tests d'usuari sobre un equip de sis documentalistes en dues fases de maduresa de l'eina.

8.2 IMPRESSIONS DELS USUARIS

A continuació es detallen les impressions que els usuaris han valorat després del seu ús amb l'eina.

- ✓ El rendiment de l'eina és notori, agrada l'agilitat amb que es poden fer cerques i prescindir de resultats a simple cop d'ull.
- ✓ L'ordre d'aparença dels assets hauria de ser seleccionable segons la data d'emissió de l'asset, o bé la data de creació de l'asset.
- ✓ Certs assets molt llargs (amb aproximadament més de cent fotogrames) tarden un cert temps en ser processats i en recuperar-ne tota la informació. Un cop recuperada, l'eina torna a mostrar la fluïdesa que la caracteritza.
- ✓ Es valora positivament que els assets es mostrin de més a menys rellevància després d'una cerca textual. Però, en canvi, el camp controlat tema/nom (no inclòs) hauria de ser tingut en compte i hauria de puntuar més que el camp de l'estrat ja que es tracta d'un camp més precís en la descripció de l'asset.
- ✓ Sobre el fet d'utilitzar eines de rànquing que empren l'estadística de les paraules (com l'eina Solr), els documentalistes comenten que en el seu procés d'indexació intenten no repetir cap paraula que ja sigui descrita o bé en el títol o bé en el resum. És a dir, les paraules repetides són vistes com a soroll en la

seva metodologia de cerca i dubten de l'eficàcia de eines que emprin aquesta solució per crear un rànquing de rellevància.

- ✓ Els usuaris també fan notar que les paraules del suggeridor de Google (emprant l'eina Google Suggestions incorporada en fases inicials de l'eina) no són rellevants donat que els texts descriptius dels assets utilitzen camps controlats. En canvi el suggeridor de paraules emprant el Thesaure de la corporació sí els és útil per a la cerca. En general, però, es pot dir que utilitzen molt el recomanador de paraules, no tant per fer-hi clic i iniciar una nova cerca, sinó per veure el context en què es troben després d'una cerca inicial.
- ✓ Per altra banda, dins dels tres possibles grups de paraules que es troben en el Thesaure (intèrprets, autors i temes), hauria de ser seleccionable quins tipus es mostren i quins no.
- ✓ Una de les opcions no disponibles però més demandades és el ressaltat de la paraula cercada en el text mostrat, sobretot en les metadades.
- ✓ Es valora molt positivament el visualitzador continu de fotogrames del Digimatge en vers el visualitzador per paginació del Digiton Digimedia. Aquest tipus de visualització els permet una navegació ràpida amb l'*scroll* i permet detectar a cop d'ull la imatge més propera al concepte que estan buscant.
- ✓ Tanmateix, els usuaris fan notar la seva disconformitat en mostrar un resum de fotogrames en un panell apartat de les metadades de l'asset. El principal motiu és el fet de perdre la referència entre l'asset i els seus fotogrames.
- ✓ Els documentalistes troben encertat el fet de mostrar ressaltats els fotogrames amb una estrella segons si contenen estrats amb la paraula clau cercada, així com poder filtrar les imatges que no contenen estrats rellevants. Veuen aquesta eina amb potencial per a una localització de moments clau dins d'assets de gran duració com sessions de parlament o events esportius.
- ✓ Quan es demana valorar la simplificació de la cerca emprant amb un sol quadre de text, els usuaris pensen que un professional necessita seleccionar sobre quins camps es fa la cerca. En cas de voler fer una cerca "a l'estil de Google", caldria fer un rànquing intel·ligent i personalitzat al tipus d'usuari de l'eina. A la pregunta de quins camps s'han d'emprar per a la cerca inicial, demanen utilitzar el camp d'indexació manual (el camp Tema/Nom).
- ✓ Els documentalistes prefereixen un menor número de resultats encertats que un gran número de resultats ambigus. Així doncs, a l'emprar dues paraules en el camp de cerca, cal fer una intersecció (AND) dels resultats enlloc d'una unió (OR) o permetre seleccionar aquesta opció a les opcions.
- ✓ Sobre la valoració de l'actual estat del recomanador d'imatges semblants. Els usuaris no troben un ús pràctic en el seu estat actual, però pensen que quan es pugui iniciar una cerca a partir d'una regió, els resultat sí seran més rellevants.

9 CONCLUSIONS I TREBALL FUTUR

9.1 CONCLUSIONS

El present treball ha demostrat que és possible integrar diferents serveis web dins d'una aplicació rica d'Internet orientada a la recuperació de vídeos a partir dels seus fotogrames. Aquest projecte té una aplicació real i directa dins el sector *broadcast* i, especialment, dins l'empresa on s'ha realitzat, la CCMA, Corporació Catalana de Mitjans Audiovisuals, una renombrada corporació audiovisual pública.

9.1.1 UN PROJECTE D'EMPRESA

El projecte s'ha plantejat i s'ha enfocat a generar estat de l'art però, al mateix temps, també a crear una eina útil que cobreixi les necessitats dels usuaris professionals d'una televisió a l'hora de cercar vídeos de manera intuïtiva. No en va, els requeriments han estat originats inicialment des d'un entorn acadèmic, però al mateix temps validats en un pla industrial. La col·laboració de varis treballadors del departament de Documentació de TV3, Televisió de Catalunya, ha estat vital per a determinar els requeriments d'usuari que han guiat la recerca.

Cal comentar que, fins on es té coneixement i a partir del treball presentat en aquest projecte, la Corporació Catalana de Mitjans Audiovisuals és pionera en el sector de la producció i emissió audiovisual; no en va ha estat la primera televisió digitalitzada a Espanya, i de les primeres televisions en tota Europa que ha implementat un sistema digital de gestió de continguts. Part de la visió d'aquesta empresa contempla integrar i viure la innovació tecnològica i de continguts com a part de la seva manera de ser, del seu "DNI"⁶¹. És per aquesta raó que s'ha acollit de molt bon grat un projecte orientat a cercar imatges basant-se en el contingut de les mateixes, per a recuperar vídeos basant-se no només en text o fotogrames, com es fa ara, sinó combinant diferents tipus de cerques de manera creativa. De fet, la CCMA reconeix que no hi ha cap altre *broadcaster* que disposi de tècniques basades en la similitud d'imatges dins els seus gestors digitals de continguts audiovisuals, cosa que permet catalogar el present projecte com a innovador.

9.1.2 OBJECTIUS COMPLERTS

El projecte tenia com a motivació principal, tal i com indicava el requisit, desenvolupar una eina per a la recuperació d'imatges a partir d'una idea. Aquest fet queda cobert

⁶¹ Tal com es pot llegir a www.ccma.cat, la Corporació Catalana de Mitjans Audiovisuals té la missió de ser un referent de qualitat i de servei públic, de ser un instrument de reconstrucció nacional i de normalització lingüística, i d'encapçalar l'evolució cap a les noves tecnologies de la informació a Catalunya.

pel projecte gràcies al mòdul recomanador de paraules. Aquest mòdul implementa recomanacions de paraules extretes del Thesaur de la Corporació, però també empra un servei web de Yahoo! Research, i el mòdul recomanador d'imatges semblants de la UPC.

Com a **requisit 2** es demanava que l'eina treballés sobre els assets de la Corporació. Aquest aspecte s'ha solucionat desenvolupant determinats mòduls en Java en l'etapa servidora de l'aplicació. El **requisit 2** també especificava que la taula d'assets que es volia abordar era la del Digition anomenat Digimedia, que no és altra cosa que una base de dades reduïda i específica per a la investigació que mimetitzava les funcionalitats de la base de dades que empen els periodistes, productors audiovisuals i documentalistes de TV3. Comentar que, un cop avançat el projecte, s'han realitzat proves que han demostrat que Digimatge i l'eina d'indexació de text Apache Solr no només són capaces d'indexar una base de dades de laboratori de la grandària del Digimedia, sinó àdhuc tota la base de dades audiovisual de la Corporació, la qual és conformada per més de dos milions d'assets. Això permet asseverar que l'eina és escalable i que és totalment factible emprar el Digimatge per atacar una base de dades audiovisual real.

El **requisit 3** presentava la necessitat que la interfície d'usuari permetés realitzar consultes ràpides via text i que, per tant, es dotés a la mateixa d'una funcionalitat d'autocompletat o suggeriment de paraules rellevants. Per tal d'assolir aquest requisit s'ha fet ús, entre d'altres, de la tecnologia Apache Solr per a la indexació de metadades dels assets.

El **requisit 4** requeria que l'eina fos accessible des de diferents sistemes operatius i diferents navegadors web. Per tal de complimentar aquest requisit s'ha desenvolupat un programa client seguint la filosofia de les anomenades aplicacions riques d'internet (RIAs) mitjançant l'entorn FLEX en la seva versió 3.

El **requisit 5** especificava la necessitat d'associar paraules text de les descripcions dels vídeos amb els fotogrames dels mateixos. Per tal d'aconseguir-ho, Digimatge implementa diferents solucions:

- ✓ Es pot cercar sobre les metadades generals d'un asset (les que apliquen a tot el vídeo) i obtenir tot seguit com a resultat els seus fotogrames. Aquesta opció té per inconvenient que no hi ha cap relació directe entre un fotograma concret i la paraula de cerca inicial.
- ✓ D'altra banda, el sistema permet associar els intervals temporals dels estrats amb el codi temporal del fotograma i, per tant, permet realitzar una cerca sobre el text que està associat a només determinats segments d'un vídeo. Aquesta opció ha estat ben valorada en el test d'usuari, però només per assets

ben detallats i que continguin moltes línies d'estrats. En canvi per a assets amb una sola línia de descripció (la major part dels assets), aquesta opció no resulta útil.

✓ Finalment, l'eina també possibilita fer cerques per exemple i permet que, a partir d'imatges sorgides d'una crida textual, se'n puguin trobar d'altres de similars mitjançant el servei web desenvolupat pel GPI de l'UPC. Aquesta tercera via de recerca es troba en constant desenvolupament, i és la més innovadora. S'espera poder portar en breu a producció -és a dir, integrar dins el gestor digital de continguts de TV3- les innovacions d'aquest projecte, i en especial de les cerques per exemple.

El sisè requeriment fa referència a la facilitat d'ús i a la intuïtivitat en la representació de varis fotogrames provinents d'assets diferents. S'han estudiat diferents tècniques i s'ha dissenyat una interfície d'usuari que permet una navegació intel·ligent, una cerca eficient i una visualització ràpida i intuïtiva.

L'estructura defensada des de l'inici del projecte ha estat la diferenciació de resultats segons si aquests eren textuais o d'imatge. Aquesta solució ha donat bons resultats, tant a nivell acadèmic com a nivell d'usuari. Presumiblement, aquest encert té bona part de raó en el fet que s'ha seguit una estructura similar al Digition Digimedia, que és per tant fàcilment assumible i entenedor pels usuaris que ja treballaven amb aquesta eina. Si es tracta de mostrar imatges d'un sol asset, l'usuari sap exactament quines imatges està veient i sobre què tracten. Ara bé, quan s'entrellacen imatges d'assets diferents que tenen en comú que provenen d'una mateixa cerca textual, els tests d'usuari han indicat que l'usuari perd sovint la referència del què estava buscant. Si a més, s'inicien diferents cerques per imatge, és probable que l'usuari es perdi completament. En qualsevol cas, aquest és un handicap habitual en la introducció de noves tecnologies, que s'espera solucionar mitjançant la millora continua del sistema, i a partir d'un lògic procés d'aprenentatge dels usuaris.

9.1.3 ALTRES RESULTATS ACONSEGUITS

D'altra banda, existeixen altres resultats col·laterals del projecte que, tot i que no s'esperaven en un inici, no per això deixen de ser gratificants un cop s'han aconseguit.

D'una banda, s'han desenvolupat un seguit de tecnologies "genèriques" (RIA, servidors REST, APIs de FLEX, etc) que s'aprofitaran dins la CCMA per a desenvolupaments de caire molt divers. Un dels requisits marcats a títol personal era aportar a l'empresa un valor afegit al treball acadèmic. És una satisfacció comprovar que aquest requisit ha estat assolit, i que a hores d'ara prototips de l'empresa, així com diferents mòduls desenvolupats en Flex, beuen directament dels coneixements provinents de la realització del Digimatge.

Una altre conclusió del projecte recau en l'elaboració d'una proposta per a la ACM (Association for Computing Machinery) SIGMM International Conference on Multimedia Information Retrieval (MIR 2010)⁶². L'edició del 2008 d'aquest congrés va tenir més de 300 propostes de les quals van ser acceptades un 18%. El mateix congrés va ser assistit per més de 150 investigadors i participants. En la última setmana d'elaboració del projecte es va notificar als investigadors que van redactar la proposta l'acceptació de la mateixa per l'edició 2010 del congrés. La proposta es pot trobar a l'Annex 1 del projecte.

9.2 TREBALL FUTUR

El projecte del Digimatge no ha consistit només en l'elaboració d'una eina RIA, sinó en assentar les bases i impulsar una arquitectura que gira sobre el concepte de servei web. Aquests passos han permès obrir diferents vies de desenvolupament en diferents àmbits del present treball, els quals tindran aplicació, tal com s'ha dit, a l'entorn de l'empresa.

Els següents punts avancen alguns dels projectes i subprojectes que derivaran de la feina del present PFC:

9.2.1 SERVIDOR

L'actual estat de l'eina està centrat en el marc del desenvolupament. Per tal de potenciar el servidor per a múltiples usuaris es pretén dotar al servidor d'una piscina de connexions (*connection pool*). Aquesta piscina mantindrà una memòria de les diferents connexions obertes a la base de dades de tal manera que es podran reutilitzar, millorant així el rendiment de les crides successives.

Es vol dotar d'un nou mecanisme de crides SQL paginades basada en la futura actualització a la versió 11 de la base de dades que utilitza la Corporació (IBM Informix). Les crides són del tipus **Select skip ? first ? * from table_name**. És a dir, es permetran dos índexs per crida, el primer (*skip*) indica a partir de quina fila es vol començar la recuperació d'informació i el segon índex (*first*) indica el número d'elements que es volen recuperar. Aquest tipus de crides permetran recuperar només un número finit de fotogrames del servidor al client. D'aquesta manera, s'evitaran petits retards en el funcionament de l'eina i evitar una de les principals fonts de problemes: el rendiment del client web.

⁶² riemann.ist.psu.edu/mir2010/

9.2.2 EINA APACHE SOLR

L'eina Solr es publicita a si mateixa com una eina empresarial. Durant la història de la seva relativament curta vida, ha anat afegint diferents mòduls de processat de text amb distintes funcionalitats. Cada mòdul requereix un estudi exhaustiu segons les necessitats de recuperació textual que es demanen per part dels usuaris. Els mòduls més interessants, per a ser inclosos en un futur al Digimatge són: el mòdul de remarcat de paraules (*Highlight*), molt demanat durant el test d'usuari, el mòdul de correcció tipogràfica (*Spell Checker*) i el mòdul propi de suggeriments (*Auto Suggest*). Cal recordar que en el test amb usuaris professionals s'ha valorat molt positivament el suggeriment de paraules similars. Així doncs, s'utilitzarà aquest últim mòdul de Solr (*Auto Suggest*) per tal de suplir el servei web de Yahoo! que, un cop acabat el projecte col·laboratiu de recerca, no pot ser emprat en un producte comercial de la Corporació.

9.2.3 EINA CLIENT

Durant la realització d'aquesta eina ha aparegut la versió 4 de FLEX, la qual aporta major rendiment i unes solucions per a la gestió d'imatges més òptimes. Un dels treballs que es realitzaran és migrar l'aplicació d'un a altre entorn de programació.

Donat que s'ha desenvolupat un servidor robust i s'ha realitzat una especificació de dades ben estructurada, resulta viable canviar la visualització de l'eina sense haver de fer grans canvis en l'estructura central. Una de les línies futures és dotar a l'eina d'una visualització conjunta dels assets amb els seus fotogrames sense que es perdi la flexibilitat actual d'unir en un mateix panell els fotogrames de diferents assets. Per fer-ho, la nova versió Flex 4 ofereix claus i possibilitats per al desenvolupament de mòduls específics des de zero basats en llistes dinàmiques enlloc de taules (l'actual component bàsic de Digimatge és més restrictiu i menys dúctil comparat amb les llistes).

Així mateix, una de les noves funcionalitats que s'integrarà és el mòdul de cerca per imatge basat en regions, enlloc de només per a tota la imatge. És a dir, podem seleccionar per exemple una part d'una imatge (un globus a una foto d'un paisatge), i cercar imatges que continguin regions a l'assenyalada. Per tal d'implementar aquest nova funcionalitat, caldrà modificar l'eina Digimatge per a atacar el servei web de la UPC o fins i tot permetre llençar una cerca amb un esbós o un color. Una altra possibilitat per a la integració de la recuperació d'imatges per regió passa per dotar l'eina FLEX de compatibilitat amb els fitxers d'arbre perceptual que descriuen les regions detectades amb l'eina GAT de la UPC. Això permetria no acotar la regió d'exemple a una selecció rectangular, sinó a seleccions més concretes que donarien lloc a resultats més acurats.

Un dels inconvenients del projecte actual ha estat comptar només amb una característica de baix nivell (el color) que no ha permès ser creatius en la visualització d'agrupacions d'imatges. Les noves versions dels mòduls de processat permetran crear

noves interfícies de l'estil del sistema cuZero o Google Swirl (eines explicades en l'estat de l'art d'aquest projecte) mitjançant el color, la textura o la forma que dotaran de més riquesa visual i conceptual a les cerques textuais inicials.

9.2.4 INTERFÍCIE PER A LA DETECCIÓ D'ENTITATS

La detecció automàtica de conceptes (cares, text a pantalla, objectes, etc) és una característica avançada amb la què s'espera dotar de valor afegit al Digimatge. La integració dels resultats de la indexació automàtica dins el Digimatge elevarà a la Corporació com a pionera del grup d'empreses del món audiovisual pel que fa a l'automatització de la indexació de continguts. Pensament que avui en dia la indexació de contingut és una tasca intensiva que requereix d'un excessiu capital humà perquè és realitza de forma manual i s'associa a un procés que requereix molt de temps. Per tant, qualsevol eina que realitzi una automatització d'aquesta funció tindrà clares implicacions tant a nivell de compte d'explotació (reducció dels costos operatius), com a nivell d'increment de la qualitat del servei (els documentalistes podran dedicar-se a tasques de major nivell i que aportin major valor⁶³, i els resultats de les cerques dels periodistes serà més precisa i rica)

En una primera aproximació, l'actual eina haurà de permetre cerques sobre els nous camps d'estrat anotats de forma automàtica, ja sigui de text, llocs, persones o objectes. El fet de permetre la cerca textual mitjançant les eines d'indexació emprades en aquest projecte serà un revulsiu per a seguir treballant en la indexació automàtica com a eina de futur. Tanmateix, val a dir que la inclusió d'aquest detector automàtic (que inicialment s'espera doni errors i falsos positius) haurà de venir necessàriament de la mà d'un sistema de validació manual d'entitats, el qual s'haurà de desenvolupar paral·lelament amb les anteriors eines.

⁶³ "Less media management, more content management"

10 ANNEXES

És presenten dos annexes al final del document:

- ✓ L'**annex 1** mostra la proposta acceptada pel congrés ACM MIR 2010
- ✓ L'**annex 2** mostra el conjunt d'escrits realitzats en el bloc d'investigació BitSearch⁶⁴ on s'ha anat descrivint el progrés del present projecte durant la seva realització.

⁶⁴ bitsearch.blogspot.com/

11 BIBLIOGRAFIA

- i **[Smeulders et al.]** Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years
- ii **[Hare et al.]** Jonathon S. Hare, Patrick A.S. Sinclair, Paul H. Lewis, Kirk Martinez, Peter G. B. Enser, and Christine J. Sandom: Bridging the Semantic Gap in Multimedia Information Retrieval
- iii **[Inoue]** Inoue, M.: Difficulties in Image Retrieval
- iv **[Salembier et al.]** Salembier, P., Manjunath, B.S., Sikora, T: Introduction to MPEG-7
- v **[Eindeberger]** Eidenberger, H: How good are the visual MPEG-7 features
- vi **[Klamma et al.]** Klamma, R., Saniol, M., Renzel D.: Community-Aware Semantic Multimedia Tagging –From Folksonomies to Commsonomies
- vii **[Cortes]** S. Cortes, Interfaz Gráfica de Usuario para la Búsqueda de Imágenes basada en Imágenes
- viii **[James et al.]** J. L. James, Z. Wang and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries.
- ix **[Cuzero]** S.-F. Chang, L. Kennedy, E. Zavesky. Columbia University Semantic Video Search Engine
- x **[Kaplan et al.]** A. Kaplan, J.Mamou, F.Gallo,B. Sznajder. Multimedia Feature Extraction in the SAPIR Project
- xi **[Giro-i-Nieto et al.]** X. Giro-i-Nieto, N. Camps, F. Marques, GAT, a Graphical Annotation Tool for semantic regions, Multimedia Tools and Applications
- xii **[Luttinger]** Luttinger A. RIA and Cloud Computing Applications
- xiii **[i3media]** MPRO, ALU, TID, MPG, CCMA, AM, TVC, NOUFER, BRAINSTORM, INFOSPEECH, FONETIC, STT. Informe Anual del proyecto i3media en el año 2007

ANNEX 1

Digimatge, a Rich Internet Application for Video Retrieval from a Multimedia Asset Management System

Xavier Giro-i-Nieto
Technical University of
Catalonia (UPC)
Barcelona
xavier.giro@upc.edu

Ramon Salla
Catalan Broadcasting
Corporation (CCMA)
Barcelona
rsalla.w@ccma.cat

Xavier Vives
Catalan Broadcasting
Corporation (CCMA)
Barcelona
xvives.t@ccma.cat

ABSTRACT

This paper describes the integration of two new services aimed at assisting into the retrieval of video content from a Multimedia Asset Manager (MAM). The first tool suggest tags after an first textual query, and the second ranks the keyframe of retrieved assets according to their visual similarity. Both applications were implemented as web services that are accessed from a Rich Internet Application via REST calls.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*information filtering, query formulation*; C.2.4 [Computer Systems Organization]: Computer-Communication Networks—*Client/server, Distributed applications*

General Terms

Management

Keywords

Tag suggestion, content-based retrieval, image databases, similarity retrieval, RIA, Web service, REST

1. INTRODUCTION

Video retrieval from archives is nowadays a challenging need for TV broadcasters. The growing amount of data daily ingested into their repositories has made no longer feasible relying only on the manual search and indexing of expert documentalists. This paper presents the integration of state of the art retrieval tools in the Multimedia Asset Manager (MAM) of the Catalan Broadcast Corporation (CCMA), a media group that produces four TV stations. The main goal of the new systems is the assist journalists in a fast retrieval of video assets to be edited and included in the video clips for the news programs.

The starting point of the presented work is Digion, the

MAM previously developed by the CCMA whose search system is based only on matching query keywords and the textual metadata manually generated by the documentalists. *Digimatge* is a new search system that adds two new services to the ones previously offered by Digion: a tag suggestion for text-based queries and a ranking of retrieved keyframes based on image similarity.

The addition of these new plug-ins into a MAM in exploitation requires a modular architecture of multiple systems interconnected with a variety of network protocols. Web services coming from the Internet offer nowadays mature technologies designed to be deployed in distributed systems, offering user experiences on the cloud as rich as if they were being run at the client side. This application are referred as Rich Internet Applications (RIA) and their technologies are also valid in the framework of a corporate network. A second contribution of the presented work is the adoption of this paradigm to extend the features of the existing MAM.

The paper is structured as follows. Section 2 presents the new services added to the existing system and how to use them from the graphical user interface. Section 3 introduces the concept of Rich Internet Application and described the system architecture and communications protocols. Finally Section 4 draws the future research lines and provides the project conclusions.

2. DESCRIPTION OF THE NEW SERVICES

The video retrieval problem is normally solved by assessing the similarity between a set of query descriptors and the target descriptors associated to the assets stored in the repository. A similarity score is computed for each comparison and a ranked list of the assets is returned to the user as a response to the query. The nature of the descriptors can be diverse and, in the case of video data, they usually represent semantic, visual and audio information.

2.1 Text and Image-based Searches

Text and image descriptors are two types of features that have been largely used for image retrieval. State of the art solutions provide examples of each case and the combination of both.

Semantic information is normally expressed under the form of text descriptors and it is the most used modality in search engines. Users can directly formulate their queries by entering keywords or full sentences to the system, and these

generated query descriptors are compared to the text or semantic descriptors previously associated to the asset. The text-to-text video search tends to provide good results as humans can be pretty precise when using text to express semantics. Nevertheless, this approach requires generating textual metadata for the multimedia content. A first option is the manual annotation of the content, a very consuming task when manually performed by a human. As an alternative, descriptors can also be automatically parsed from the image filenames or contextual text, or be generated through signal processing like OCR-based solutions [5] or semantic classifiers [9].

A second family of retrieval systems are based only on visual descriptors, low-level features automatically generated by applying signal processing algorithms on the content. The automatic nature of the process allows processing a larger amount of content and the generated descriptors are as accurate and complete as the set of feature extractors. This solution, though, presents two drawbacks when compared to text annotation. Firstly, the correlation between perceptual descriptors and semantic concepts is not as close as in the textual case and, secondly, the user needs to become familiar with a new interface that formulates queries in terms of perceptual descriptors instead of text. There are mainly two options for the user to formulate the query, whether by directly providing a quantified value of the descriptor or by providing the system some sort of visual content from which the query descriptors can be extracted, such as an example or a sketch [1] [3] [4].

Although the visual-based similarity provides reasonably good results from a semantic point of view, most system exploiting these technologies also rely on textual descriptors, providing hybrid search solutions that combine both. Fast indexing and retrieval algorithms are more mature in the text mode than in the visual, so many solutions use text queries to retrieve a first set of results and these are later refined by a second search based on visual descriptors. This third family of retrieval systems are the natural evolution of the two previous, introducing multimodality to the search experience. Figure 1 shows an example of combined text and visual query, in which the semantic class of the object *dog* is expressed through text, but the color of its hair is represented through a palette choice. Commercial examples like Google Similar Images, Pixolu or Xcavator have applied this strategy.

2.2 Proposed solution

The presented system also combines text and visual descriptors in the framework of a broadcaster MAM. Textual annotations are generated by a team of expert documentalists according to some guidelines and recommendations. The manual annotations can refer to two different scales of a video asset: a global or local scale. Global annotations describe the whole shot as, for example, "Soccer game: F.C.Barcelona 3 - Athletic Bilbao 1". while local annotation are limited to the segment between to two time codes, as it would be "Player X scores a goal". The textual descriptors are complemented with new visual descriptors automatically computed from the video keyframes. Using keyframes instead of full videos reduces the amount of data generated during indexing and that needs to be processed at search time. In most cases,

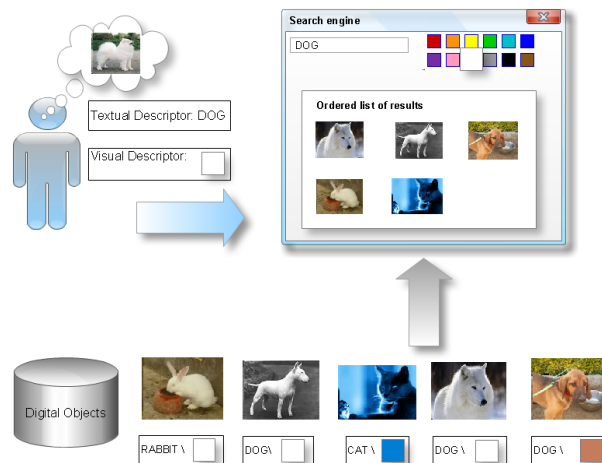


Figure 1: Query based on text and visual cues.

a few keyframes of a video sequence are enough to effectively represent its semantic content. The system computes four MPEG-7 visual descriptors from each keyframe: Color Layout, Color Structure, Edge Histogram and Texture Homogeneous.

The present work focuses on a subset of perceptual descriptors, those computed on still images. Although the final goal of the system is the retrieval of videos, the evaluation of the similarity between queries and target content is performed at the keyframe level. This approach is taken to reduce the amount of data generated during indexing and that needs to be processed at search time. In most cases, a few keyframes of a video sequence are enough to effectively represent its semantic content.

The search process begins with the input of a textual query in a box located at the upper left side of the graphic interface. While typing the query, an autocompletion combobox appears suggesting terms from a thesaurus maintained by the documentalists. This tool speeds up the typing process as well as decreases the probability of spelling errors. The words in the thesaurus are preferably used by documentalists, but during asset metadata may contain many more words are used. For this reason, the user can also enter keywords that are not included in the thesaurus.

Once the textual query is introduced, the search is executed and results are displayed in the graphic interface shown in Figure 2. The upper part of the screen shows a table with the numeric ID of the retrieved video assets as well as their title. The lower half contains the ten first keyframes of each retrieved video object. A double-click on any row in the results table repaints both upper and lower parts of the interface with more data about the selected video object: the results table is replaced by a all textual metadata associated to the asset, while at the lower part a new tab is created to show all keyframes from the selected video object.

A new tag suggestion service [2] is also executed at query time to retrieve similar terms to the entered keyword. This service is based on a statistic analysis of terms in the textual

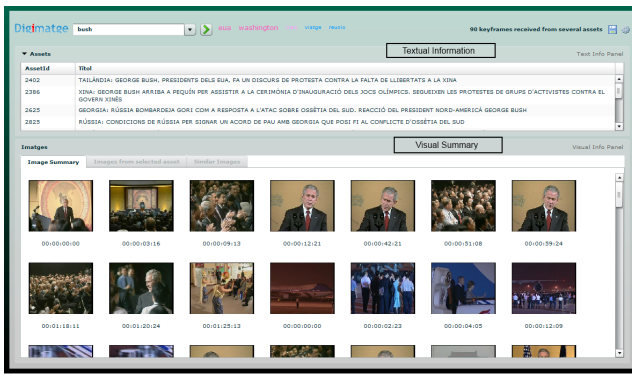


Figure 2: Screenshot of the graphical user interface.

metadata and shows its results in a word cloud. By doing so, the system is proposing new query terms aimed at assisting the user when its search idea is vague or fuzzy.

In addition, a new visual search service is available by double clicking on any thumbnail. When doing so, an image similarity search is assessed between the selected keyframe and the rest of the keyframes of the retrieved assets. The process can be understood as a reranking of keyframes, but in this case, all keyframes of the retrieved assets are considered, instead of the ten first keyframes that are shown after the initial text query. The results are displayed in a new tab, where any click on a keyframe will show the asset summary on the upper part of the interface.

3. IMPLEMENTATION

3.1 Web Services and Rich Internet Applications

Video content in the broadcasting industry is normally stored in Multimedia Asset Managers (MAMs), a distributed system to access a content through queries on an indexed database or catalog navigation. The addition of new services to an existing system is an operation that must face several technical restrictions due to the existing installations but, at the same time, offers the opportunity to introduce state of the art technologies. In the presented work, the distributed architecture of the Internet inspired the chosen approach to access the new services through a Rich Internet Application (RIA).

RIAs [8] are a conceptual framework for developing applications based in two basic trends of computing during the last years. In one hand, applications run on local computers search for information sources on distributed remote servers; on the other hand, many applications that used to run at the client-side are now accessed on a remote server through a web browser. Moving local resources and computations to remote servers is has been referred to cloud computing, while those applications and data sources available on the cloud are referred as web services.

The term of RIA (Rich Internet Application) is applied to a piece of software that is accessed from the Internet but whose operations are distributed between the local machine and remote servers. Although RIAs are in many cases ac-

cessed through a plug-in installed on a web browser, their possibilities and efficiency are higher than for a website. The design of a traditional web site is based on answering to a user action, forcing an access to the server after every new event and waiting from its answer for a refresh. Alternatively, RIAs are more data driven applications. They wait for the user to prepare all required data before connecting to the server. This approach is possible thanks to the graphic elements being persistent after the first download.

RIAs at the client side can be implemented in multiple programming languages and development environments, also called frameworks. The first RIAs were implemented in Javascript and AJAX, and take advantage of the native adoption of Javascript support from most web browsers. However, this open approach results in certain cases in multiple implementations that make this option less stable than other closed solutions as Adobe Flex or Microsoft Silverlight. These two proprietary technologies use, respectively, Action Script and .NET as programming languages. The third version of Flex framework uses the AS3 and MXML languages and is supported by most web browsers through the Adobe Flash Plug-in; while Microsoft Silverlight is based on .NET language and its middleware is shipped from fabric since Microsoft Windows 7. Java is another important language on the web and its compiled classes can be easily downloaded and safely executed on the client over the Java Web Start framework. By definition, RIAs must communicate with services running on remote machines, and these servers must respond to requests more complex than just sending static web pages. For this reason, it is also necessary to use at the server side languages that can keep the state, such as Java or .NET, or interpreted languages, like PHP or ASP.

The data exchange between web services and RIAs is achieved over a certain communication protocol that must be capable of transporting amounts of data significantly higher than in traditional websites. A first family of protocols has evolved over the HTTP layer, as in the cases of SOAP [6] and REST [7]. SOAP uses XML files in every communication layer and it is an official standard by the W3C, while REST is a simplification of it that uses lighter messages. The overhead introduced by the XML-based messages in SOAP is removed in REST, and its implementation simplified by using the same GET/PUT native functions included in HTTP. A common practise in REST communications is to answer the REST requests with JSON, a lightweight data-interchange format based on JavaScript. A second family of communication protocols for RIAs are not based on HTTP. For example, the open sourced AMF (Action Message Format). This protocol allows RIAs implemented in Flex to use remote classes implemented in Java, AS3 and other several server languages.

3.2 System Architecture

The proposed architecture combines several technologies that exploit the advantages presented in the previous paragraphs. The main challenge in the design is to integrate heterogeneous different information sources in a modular approach that could be expanded in the future if new services are to be added in the future. Figure 3 shows and overview of the complete system architecture. A main server (1) connects the different resources located inside the corporate intranet or outside through the public Internet.

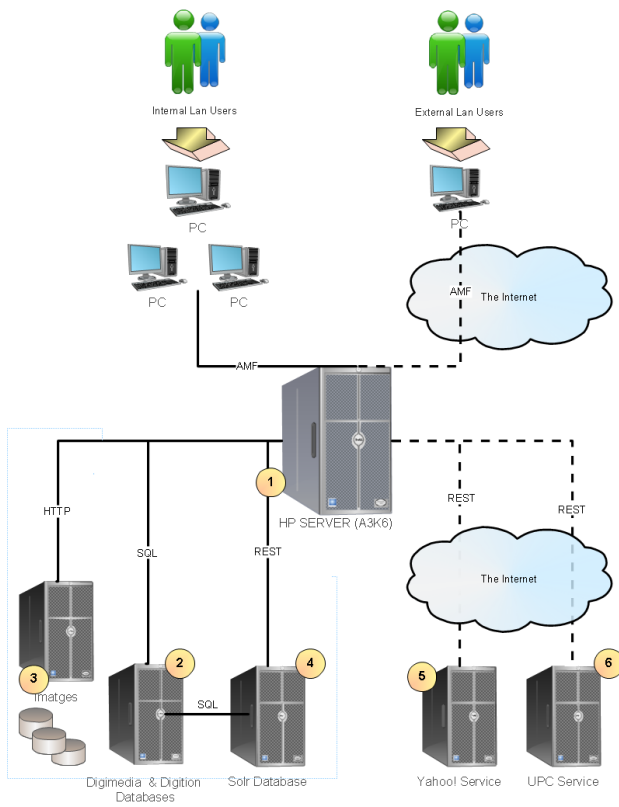


Figure 3: Systems elements and their connections.

Apart from connectivity, the main server (1) is the responsible of two other tasks. Firstly, it processes the text-based queries and retrieve the IDs of the resulting video assets. Secondly, it generates the standard definition keyframes from the time codes defined by the keyframe extractor. During the first tests on the system, it was detected that the text-based queries on the Informix database (2) could not be answered as quickly as required through classic *Select* queries to the SQL tables describing the 454,626 test assets. As a solution, a new platform capable of efficiently index and search text information was introduced (4) based on *Solr*, an open source enterprise search platform from the Apache Lucene project. The communication with this text-based search engine is established from the main server over REST. The second task controlled from the main server is the generation of keyframes from the time codes stored in the Informix database. These time codes are generated at ingest time by the keyframe detector. The extraction of keyframes requires accessing the time code database as well as the content repository to retrieve to actual video. Keyframes are generated with *ffmpeg*, a free software tool for video coding, and stored in host (3).

The two remaining elements in the architecture are those providing new features to the system: the tag suggester (5) and image similarity (6) web services developed by Yahoo! and the Technical University of Catalonia (UPC), respectively. Both of them are accessed over REST, but the text suggester returns an XML data structure and the image ranker directly provides a JSON object for easier integra-

tion.

Finally, the Flex RIA is downloaded from the main server and can be executed on a web-browser with a Flash plug-in or as a stand-alone application over the Adobe Air middleware. The communication between client and server is over the AMF protocol, using the BlazeDS implementation. This option serializes Java and Flex objects at the two sides of the link.

4. CONCLUSIONS

This paper has presented the implementation of two additional video retrieval services in an existing MAM. Their integration in the existing system required the definition of a system in which data and applications are distributed in heterogeneous servers running different technologies. The access to the new services was implemented as a RIA, allowing simple access to all users through a conventional web browser. The additional services running in external hosts were successfully connected to the corporate systems, providing guidelines for the future deployment of similar extensions. After these first successful steps, the system will continue evolving to introduce new methods for results clustering and navigation, efficient indexing for visual descriptors and connectivity to external news sources.

5. ACKNOWLEDGMENTS

This work was partially founded by the Catalan Broadcasting Corporation (CCMA) and Mediapro through the Spanish project CENIT-2007-1012 i3media.

6. REFERENCES

- [1] M. Flickner and al. Query by image and video content: The qbic system. *Readings in multimedia computing and networking*, pages 255–264, 2002.
- [2] N. Garg and I. Weber. Personalized tag suggestion for flickr. In *WWW 2008 Conference Proceedings*, pages 1063–1064. ACM, 1997.
- [3] J. L. James Z. Wang and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(9):947–963, September 2001.
- [4] R. Kumar and S.-F. Chang. Image retrieval with sketches and compositions. In *ICME 2000 Conference Proceedings*, pages 84–89. IEEE, August 2000.
- [5] R. Lienhart. Automatic text recognition for video indexing. In *ACM Multimedia 1997 Conference Proceedings*, pages 11–20. ACM, 1997.
- [6] E. Newcomer. *Understanding Web Services: XML, WSDL, SOAP and UDDI*. Pearson Education Inc., Boston, Massachusetts, 2002.
- [7] L. Richardson and S. Ruby. *RESTful Web Services*. O’Reilly Media Inc, Sebastopol, California, 2007.
- [8] D. Woolston. *Pro AJAX and the .NET 2.0 platforms: Chap.5 Rich Internet Applications*. Springer-Verlag, New York, NY, 2006.
- [9] J. Y. Yu-Gang Jiang, Chong-Wah Ngo. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR 2007 Conference Proceedings*, pages 494–501. ACM, Jul 2007.

ANNEX 2

Bit search

This blog describes the research projects directed by Xavier Giró i Nieto, an assistant professor from the Image Processing Group at the Technical University of Catalonia (UPC). It is mainly written by its students and Xavi himself to show the evolution and thoughts about their research.

Showing posts with label **RAMON**. [Show all posts](#)

Thursday, September 24, 2009

true intelligence?

I watched a video some weeks ago about Concept Recognition by a robot, a topic related to image retrieval (the one I'm doing research for my thesis), and i didn't realized what important is the image retrieval for Human Being.

In the video, we can see Asimo, the Honda's Robot trying to do something we easily do without knowing even how. Recognizing sizes, forms, colors, shapes as descriptors of a whole bunch of generic objects (a bicycle, a ball, a table, a house).

At the end of the video, a chair will be show to Asimo and he(it?) will be asked to decide whether the object is a chair or a table.

For me that is the beginning for some sort of **judgment**

Posted by Ramon at [10:47 AM](#) [0 comments](#) 

Labels: [identification](#), [image](#), [RAMON](#)

Sunday, September 13, 2009

Being in the Cloud

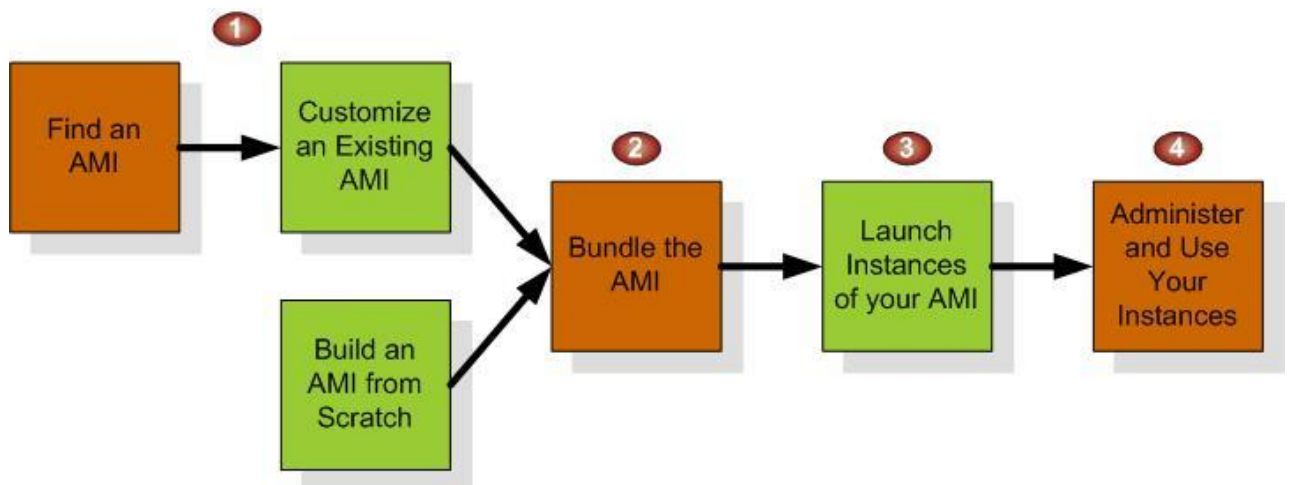


Cloud Computing is like Web 2.0: there are lots of definitions, difficult to explain and understand, so at the end all of them are accepted.

In plain words. Are you running out of memory in your hard drive? rent one. Are you mad of your HP 2 GHz Server Core for hosting your professional web page? rent a bigger one online and put your server in. So you are an enterprise (or just a little one) and you need software on a Windows operating system plus a Linux one plus a Leopard and using the output of one application as the input of another is quite a nightmare? well, virtualize them all in the Internet.

That's Cloud Computing. To upload all your resources on the line. Broad band will be appreciated! These are some good renting plans to start:

[Amazon Elastic Computer](#) it's the first in its class with a plethora of Data Bases (oracle, ibm, mysql, sql server even postgresql). A basic AMI (Amazon Machine Image) with 160 GB storage, will cost to you \$0.10 per instance hour. If your web is about sport and soccer and your public is spanish you will expect to increase your demand, that's why with just 2 or 3 clicks you can extend your instance to a 64 bit platform with 8 cores. You even can upgrade just the core , the RAM memory or the instance storage. The next figure is from [Amazon Developer Site](#):



[Windows Azure](#) is quite the same as Amazon's but with a very good marketing team in spite of its focused idea for running just applications made with it's own Microsoft .NET Sdks. Therefore, no play for you Java programmer.

Maybe you will have better luck with [Google App Engine](#) that it let you "run your own applications in the Google infraestructure". You can start developing for free with your python or java knowledge. I have tried it and it's as easy as download their Eclipse plugin, make your web application and deploy your WAR online.

All of three possibilities will allow SQL sentences to access, retrieve and manage your DB data. Rest services to communicate with another online services, etc...

[Some](#) people says C.C. is just a revolution in the way we understand Operating Systems, desktop applications and connectivity. Others (whose target is compromised) just say its enthustiathsm will decrease. Who? [Quite easy.](#)

Posted by Ramon at [8:44 PM](#) [0 comments](#) 

Labels: [Cloud Computing](#), [RAMON](#), [rest](#), [webservices](#)

Sunday, June 28, 2009

Image and Text queries

Hi there.

In my previous posts, I talked about some technologies to create an image search application based in text queries or image queries, it doesn't matter which.

After this long time studying how to create a program that was accessible and easy to use, I resolved to learn programming what it is called a RIA (rich internet application) based on Flex from Adobe.

This "search" application main purpose is to let the documentalists from CCMA (Catalan Broadcasting Company) dive into their large database of images and videos and retrieve information that was not previously entered as text. So, to let this clear, a video without any metadata text attached to it, it's completely lost. The actual state of the art of DAM applications (Digital Asset Managers) is just to let you retrieve your assets by text queries.

For instance, what the GPI department of UPC is trying to solve is to retrieve images similar to other images. And I want to contribute by encapsulating this technology with a useful and live database with lots of images with text manually entered by professional documentalists.

Therefore, if a documentalist wants to get a picture of Obama with a cap to trigger a new search, he or she would search directly "Obama" using a text query. But likely more than hundreds of Obamas will show up. So, we could manage to enrich this text query with an image query of a cap. The documentalist will find a particular image of Obama with a cap which could be useful to attach to an article with a title like this "Obama likes sport"

What is necessary in this flow is to have an image representation of a cap. This could be achieve launching a text search to a web service like Flickr with the text "cap". It could also be needed to find an appropriate word to deliver an exact match for an image. We could use a web service like [Google Suggest](#) for take this most suitable word. By the way, this kind of application enriched with web services is called a

[mashup](#).

More ambitious is the idea of transform a query like:

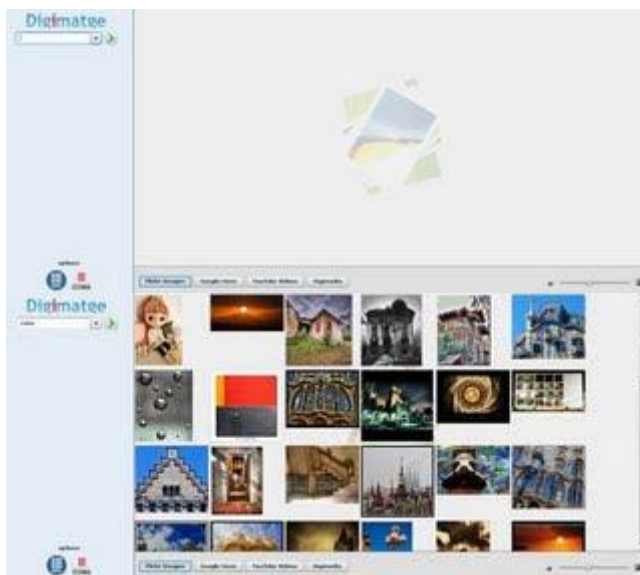
" Obama, suited in a black jacket, wearing a tie "


into...



I have taken a shot from the current state of the tool. It currently allows you to do simple text queries to some open apis and most interesting, it allows to do some initial query with images with just a predefined dataset established between the UPC and the CCMA.

The next step will be to think a way to use text and image queries to filter the results and enclose the best image representation of the starting key idea.



Posted by Ramon at [10:14 PM](#) [0 comments](#) 

Labels: [query by image](#), [query by text](#), [RAMON](#)

Monday, May 18, 2009

Project presentation at i3pfc



Hi,

The week before I presented some work in progress about my PFC in the I3pfc session at the [UPC \(Universitat Politècnica de Catalunya\)](#) in Barcelona.

The topic was RIAs and webservices content delivery.

The presentation was an opportunity to have knowledge about my own self-assurance in front of the audience. Furthermore, some lessons learned are:

- * The amount of information must be trimmed to not saturate the audience.
- * Some text could be enclosed to each slide to point out the main ideas behind each one.
- * Not use technical words without previous definition.
- * Be accurate and polite as the importance of the final projecte deserves it.

So, I am both nervous and excited to prepare the new presentation focused on the final project delivery.

Posted by Ramon at [10:59 AM](#) [0 comments](#) 

Labels: [RAMON](#), [webservices](#)

Saturday, May 2, 2009

YouTube Nightmare

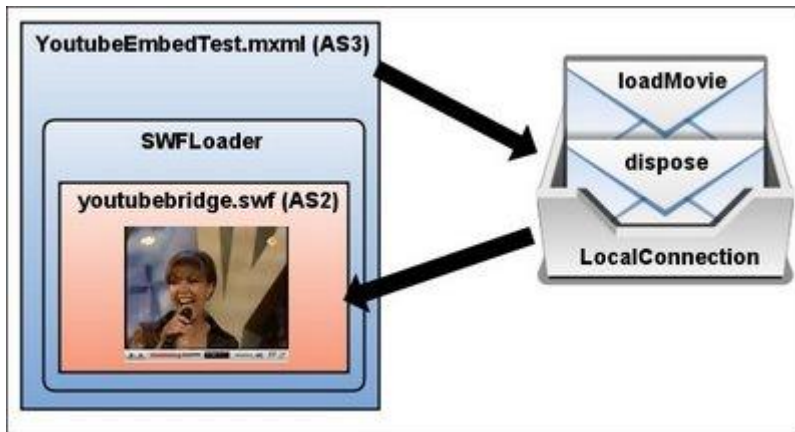


These are good steps for those who are suffering the difficulties for adding videos on Flex 3 applications:

You have to define whether your video is an flv (or adobe flash suitable media) or it is a youtube video. In the second case you must use the youtube player and use the **uid** from youtube video url to play it. Other tricks like [getting the flv url from YouTube service](#) will not work (as youtube could change the system) and are not legal.

I know, you have searched a lot in google and you have found lots of tutorials for rip the flv url and to play it in a custom flex player like [this excellent player](#) from fcomponents . That's because adding the youtube official player on your flex 3 application is quite a nightmare as it was coded in Action Script 2. And Flex 3 uses mainly as3.

The best way to add the YouTube player is to use [TubeLoc](#) which works as a wrapper using the Loader class.



To use it, instantiate the **videoplayer.as** component and add it as a child within some component and set its **videoid** by its setter method. The setter method will launch the **loadVideoById** that would start the youtube video loading by loading the as2 swf bridge too.

To avoid **netconnection** issues, instantiate only one videoplayer class. This is a caveat from tubeloc. As the same tubeloc autor says: "You can't have two YouTube players on the stage at the same time".

Accessing to youtube videos is quite easy with [YouTube api](#). I used the parameter **&alt=json** and parsed the result from my HTTPService Call.

Code

...

```
var jsonObj:Object = JSON.decode(event.result);
var entries:Array = jsonObj.feed.entry;

for (var str:String in entries)
{
var video:AssetVO = new AssetVO();
video.videoUrl=entries[str].media$group.media$player["0"].url;
video.videoId = video.videoUrl.substring(video.videoUrl.indexOf("?v=")+3);
video.thumbUrl=entries[str].media$group.media$thumbnail["0"].url;
}
}
```

...

Take note I kept both videoUrl and videoId as the parameter to pass to TubeLoc. Also note I took the first thumbnail from the 3 YouTube let me use to represent the video.

Another problem will occur when you upload your swf main application to your server. As Adobe Security Policy System will not allow accessing data from other servers but the server where the swf is allocated (I know this is quite difficult to understand. Just remember, every HttpService must have its file crossdomain.xml added somewhere in your flex code).

In my application I added this lines in the preinit() function in order to download images from YouTube service.

Code

...

```
private function preinit():void
{
Security.loadPolicyFile("http://i.ytimg.com/crossdomain.xml");
Security.loadPolicyFile("http://s.ytimg.com/crossdomain.xml");
Security.loadPolicyFile("http://gdata.youtube.com/crossdomain.xml");
Security.loadPolicyFile("http://www.youtube.com/crossdomain.xml");
}
```

Posted by Ramon at [11:54 AM](#) [0 comments](#) 

Labels: [ACTIONSRIPT](#), [RAMON](#), [webservices](#)

Sunday, April 19, 2009

More about rest, json and flickr.

RestFul Systems relay on the well know http protocol as comunication layer. That is using the *Get*, *Post*, *Put* (**1**) and *Delete* methods inherited from http.

As i wrote in my previous post, rest calls are "inside" an http call leveraging all the xml stuff necessary in Soap. So, any web browser can communicate with a Rest System in a breeze.

In my Project I needed to talk with both google and flickr apis. They serve for different purposes but for images, flickr win. Why? basically it let you retrieve 100 images per call. Google does 8!

There are lots of methods for flickr api, some use get and some us post. For example, a Get method could be the *flickr.photos.search* method. Each method has its own "manual" page. So, take a look at

<http://www.flickr.com/services/api/flickr.photos.search.html> and each method also has a api explorer

<http://www.flickr.com/services/api/explore/?method=flickr.photos.search>. Try just to fill the **text** tag box and press enter button in the api explorer.



The api explorer let you only get the results in xml, but you can also get your results in: json or php. Just add another tag to you html query like [&format=json](#).

For example, searching the text "tarragona" (by the way, one of the most beautiful cities in the world):

[Press for xml results](#)

[Press for json results](#)

[Press for php results](#)

Note that the results are photo ids and not the urls, you must build your urls in order to watch them (3).

Like [Manel](#), I also prefer to work with Json. It is the fat-free alternative to xml. Some Pros and Cons in (4). With [this Flex](#) application you can run some benchmarking between them. That is the result of importing 500 data rows with soap, xml, json, and amf (amf will be described soon in this blog). And the results are (look both the bandwidth and the time consumption). Json is a winner for a tippically text data binding approach.



Finally, if you want to build your own Java Rest Server as im doing right now. Take a look at the [Restlet Framework](#). A very impressive framework to build both server and client sides of a rest communication in java.

(1) Post and Put differences? (I always wanted to know that)


<http://www.elharo.com/blog/software-development/web-development/2005/12/08/post-vs-put/>

(2) Get a Flickr Api Key. <http://www.flickr.com/services/api/keys/>

(3) Build the photo urls. <http://www.flickr.com/services/api/misc.urls.html>

(4) json vs xml <http://www.json.org/xml.html>

PD. for those who are wondering how to pronounce JSON. It is like Jason ("geison"). But i will continue to say ("jotason"), the catalan way ;).

Posted by Ramon at [12:05 PM](#) [0 comments](#) 

Labels: [FLEX](#), [RAMON](#), [RIA](#), [webservices](#)

Sunday, April 5, 2009

Web Services Introduction. Soap and Rest protocols.

SOAP is a protocol which aim is to provide an accesible way to call remote services (typically code functions) by only using **xml**'s in the process. All the system is described by another xml called **wsdl**. The goddness of using only xm'ls in the **transaction and definition** of the service is that xml is actually supported by all progamming languages. So, SOAP is **interoperable** between java, flex, .net, c, python...



But, why is interoperability so important in the business context?

Let's put an example. A company wants to deliver some weather information coded with .net. Each call is well paid by a TV company, for instance. But, this company's programmer just know java.

What a pitty for both companies!

Luckily, SOAP bypasses this issue. Let's see the xmls involved in this anotAnother example could be like this:

XML SENT BY THE CLIENT

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock"> // where ?
<m:GetStockPrice> //what function ?
<m:StockName>IBM</m:StockName> //what parameter ?
</m:GetStockPrice>
</soap:Body>

</soap:Envelope>
```

Translation: Hey Server, give me the stock price of the company IBM, these are hard times.

XML SENT BY THE SERVER

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
<m:GetStockPriceResponse>
<m:Price>34.5</m:Price>
</m:GetStockPriceResponse>
</soap:Body>

</soap:Envelope>
```

Translation: Hey Client, take this. Good luck with your investments!

So, we have implemented a good way to talk between companies using a web service. And we have only relied on XML to understand each other. But, for just retrieving an "int" there is a lot of **verbosity** in this xml-conversation. And that is one of the most complaining issues of SOAP.

After all, this verbosity (all this xml encapsulation text) is necessary to make this conversation [B2B](#) oriented and [w3c](#) compliant.

That's why Google has deprecated its [SOAP Search Service](#) by a more user-friendly [REST service](#). REST is **not** w3c compliant but its quite simple as it just rely on **HTTP** get & post features to communicate.

So yes, you can talk with google webservice with just using your explo... i mean your firefox.

A quick example (click on the link below and see the result):

<http://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=Paris%20Hilton>



Paris Hilton ?????!!!! hey, i can hear you buzzing out there. Be quite, because that's the **official** Google example for its service ;)

As you can see, the returned object is a **JSON** compliant object (Java Script Object Notation). So it is simply an **object** with some **tags and values**. You would use one of these four **url** variables which Google brought to you.

... "url": "http://es.wikipedia.org/wiki/Paris_Hilton" ...

Google also have an image web service in REST, lets find some Zapatero faces.

<http://ajax.googleapis.com/ajax/services/search/images?v=1.0&q=zapatero&imgtype=face>

Look at the first result:

... "url": "http://blogs.librodearena.com/myfiles/hayek/20071127171523-zapatero.jpg"

...



But I really think he is quite a good president.

More info about **SOAP** and **REST** and **WebServices** in general in [this fantastic article](#) by xml.com.

Posted by Ramon at [9:42 PM](#) [0 comments](#) 

Labels: [RAMON](#), [webservices](#)

Saturday, March 28, 2009

Hello World

Hi there!

In my [first post](#) I presented Flex and what a RIA application is.

In this post, I want to scarcely show you both [mxml](#) and [as3](#) languages which they perform (with xml and css too) the greatness of flex.

It is not my aim to write more posts talking about coding, there are lots of excelent tutorials (like the [live docs](#) from adobe) in the net. But I want to show how easy it is to design a flex application.

Flex tries to separate the work between inspire designers and hardcore programmers. So, in Flex, you would start with an **mxml** file (like a java file with the main). Mxml is like xml and its a very descriptive language, so it starts with:

```
<?xml version="1.0"?>
```

As it is the main program, the next row will tell to Flex compiler that this is the main application:

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" width="100%" height="100%" layout="absolute">
```

Let's add some Graphic User nodes to our Hello World classic app. Note that each node is absolutely positioned inside a Panel Container. I have added a Text, a Text Input and a Button.

Note also that the button fires a **Greet();** function when the user clicks on it.

```
<mx:Panel horizontalCenter="0" verticalCenter="-156" title="first App">

<mx:Text id = "mytext" x="10" y="10" text="Your Name:" fontWeight="normal"
fontSize="11"/>
<mx:TextInput id = "mytextinput" x="86" y="8"/>
<mx:Button id = "mybutton" x="86" y="38" label="Greet Me!" click="Greet();"/>

</mx:Panel>
```

Let's code the **Greet();** function with AS3, this is an ECMA Script compliant code (like javascript). Therefore, if you are a javascript coder, You almost are an AS3 programmer.

AS3 code could be inside the mxml file within the tag **Script** or outside the mxml file in a ".as" file (for a cleaner code structure). Let's take the first option for this little app.

```
<mx:Script>

<![CDATA[
import mx.controls.Alert;

public function Greet():void{
if (mytextinput.text=="")
{
Alert.show("text here");
}else
{
Alert.show("Another text here");
}

}

}]>

</mx:Script>
```

You can Compile this. Generate an SWF file, import it in a HTML file and upload them into any common server.

[SEE EXAMPLE 1](#)




Ok, but, how can the designer help in this? Quite easy, look at this [skin.css](#) file of the second example. And now, run it:

[SEE EXAMPLE 2](#)



Better, don't you think?.

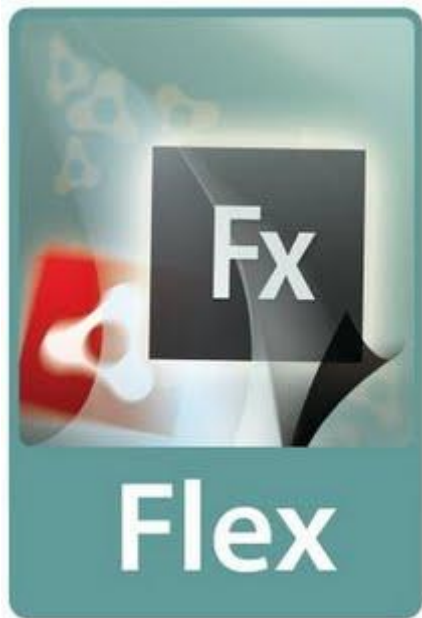
You can download all the project source by doing a right-mouse-click on the second example and click "view source".

Posted by Ramon at [12:34 PM](#) [0 comments](#) 

Labels: [ACTIONSCRIPT](#), [FLEX](#), [RAMON](#)

Saturday, March 21, 2009

Flex



Hi there!

I am Ramon Salla, student from Etsetb (UPC, Barcelona). I am also working as senior project manager at [CCMA](#) (leader broadcast company from Catalonia).

My project, which is being lead by Xavier Giró (UPC) and Xavier Vives (CCMA) , consist in the development of a graphic interface, accessible from the web, which should be useful to reporters, archivers and journalists from CCMA to test and give feedback on some of the new cool tools developed by UPC (GPI) in the context of the spanish funded [i3media](#) project.

So, my aim in this blog is to share with you my thoughts about coding a RIA (rich internet application) software. That is why in this first blog i will try to summarize what a RIA is.

A RIA app is a **Web-based** application that approaches the **speed** and **elegance** of a local application. It could be coded using any code language that a web browser understand (maybe via plugin). A very well-know ria apps are:

[Gmail](#) (JavaScript + Ajax + ...)

What do you need to run it ? nothing

[PhotoShop Express](#) (AS3 + mxml + Flex),

What do you need to run it ? [Adobe Flash Player](#). If you can see youtube videos, you already have it ;)

[Silverlight Showcase](#) (JavaScript + .net + c#)
What do you need to run it ? [Silverlight plugin](#)

JavaFX (Beta) (Java)

You could find a really (really!) great performance test [here](#).
Another great page is [RiaStats](#).

Futhermore, I have decided to develop my RIA with Flex. Why?

because of its simplicity over other codes

because of its impressive documentation and lots of programmers out there.

SDK is open source (not the builder which is based on eclipse).


because of BlazeDS based on adobe's AMF open source protocol to serialize objects between java server an flex client (more in antoher post)

because of its spreaded adobe flash player (even in portable handsets, iphone, winMO, android...) . And because it runs on mac, win and linux whereas microsoft silverlight doesn't.

because of its flex brother called AIR which ports all your web flex apps easily to the desktop (more in another post)

because adobe is orienting flex to the business. Read [this interview](#) for more info.

That's all, I hope i could show you some code examples by the next post.

Posted by Ramon at [11:21 AM](#) [0 comments](#) 

Labels: [FLEX](#), [I3MEDIA](#), [RAMON](#), [RIA](#)