

Utilizarea tehnicilor de sincronizare a interacțiunii pentru implementarea unui sistem hypervideo

Marian Dârdală, Adriana Reveiu, Titus Felix Furtună

Academia de Studii Economice – ASE București

Piața Romană, Nr. 6, 010374, București

E-mail: dardala@ase.ro, reveiua@ase.ro, titus@ase.ro

Rezumat. Aplicațiile multimedia utilizează tehnici speciale pentru parcurgerea propriului lor conținut, cum ar fi hypertext și hypermedia. Aceste două concepte oferă utilizatorului alternative semantice în accesarea unui document multimedia. Un mod particular de concretizare a conceptului de hypermedia este hypervideo. Principala dificultate în realizarea unui astfel de sistem se referă la faptul că, data video are, pe lângă caracterul spațial și dimensiune temporală, adică ea se derulează în timp. Implementarea unui sistem hypervideo presupune definirea mai multor module, specializate pe tipuri de operații și care urmează să conlucreze între ele. Astfel, în articol, se prezintă o structură de date, adaptată, pentru a gestiona partea de sincronizare a evenimentelor de interacțiune cu utilizatorul, în concordanță cu momentul derulării secvenței și cu obiectul din cadrul vizualizat, care a recepționat evenimentul. Pe lângă structura de date propriu-zisă sunt prezentate și operații uzuale de întreținere a ei cât și de exploatare în contextul utilizării sistemelor hypervideo.

Cuvinte cheie: hypertext, hypermedia, hypervideo, arbore de segmentare cadre, flux video, sincronizarea interacțiunii.

1. Introducere

Implementarea unui sistem hypervideo are particularități derivate în mod deosebit din natura datei care furnizează legătura.

Dacă în cazul conceptului de hypertext legătura este de natură statică și se reflectă printr-un text, mecanismul de control al legăturii se complică pentru date dinamice care se derulează în timp, cum este și cazul datei de tip video.

Se propune astfel, definirea unei structuri de date care să fie capabilă să gestioneze atât informații cu privire la persistența în timp a obiectelor din fluxul video cât și informații ce localizează diferitele obiecte în cadrele în care apar.

Pe de altă parte, ferestrei de vizualizare a secvenței video trebuie să i se asocieze evenimente de interacțiune cu utilizatorul care să se poată declanșa doar în anumite momente și locații.

Un alt aspect, deosebit de important, pentru sistemul hypervideo se referă la modul în care se prezintă informația referită. Aceasta poate fi inclusă în prezentarea video sub forma unei ferestre suprapuse peste o parte a cadrului video și care să apară concomitent cu derularea secvenței și să persiste până când este închisă, explicit, de utilizator sau se poate furniza, ca legătură, un URL la care să se facă trimitere. În acest mod, se impune oprirea derulării secvenței când s-a recepționat evenimentul *click*, se deschide pagina aferentă URL-ului și după consultarea ei, utilizatorul poate continua cu derularea secvenței video de unde a rămas. În acest fel, utilizatorul nu pierde șirul prezentării video, ci accesează un conținut informațional în scopul clarificării unor aspecte, concepte prezente în secvență.

Tot prin mecanismul de sincronizare este posibilă accesarea unei entități informaționale de tip video pornind de la un element de tip hypertext sau hypermedia. Spre exemplificare, dacă într-o prezentare există o imagine ce reprezintă un cadru dintr-o secvență video sau se dorește accesarea unei subsecvențe dintr-o secvență video care are legătură cu imaginea statică din prezentare, este posibilă tratarea acesteia ca element de tip hypermedia și prin ea să se facă trimiterea către o secvență video ce se va derula începând cu un anumit cadru. În acest caz, secvența video joacă rol de element referit și nu necesită elemente suplimentare de sincronizare, ci este nevoie de un player video capabil de a derula subsecvențe identificate prin cadrul lor de început, eventual prin cel de sfârșit.

În toate cazurile prezentate, mecanismele de sincronizare se bazează în primul rând pe accesul direct la secvențele video digitale.

2. Accesul direct la secvențele video

Player-ele video pot accesa în mod direct o subsecvență video sau chiar un cadru individual din fluxul de imagini. Accesul direct la diferitele cadre ale unui film presupune indexarea, invariabilă în timp, a imaginilor fixe ce compun mișcarea. Acest lucru se realizează cu ajutorul sistemelor de reperaj cunoscute sub denumirea de *Time Code* și de *Frame Code* (Smeureanu & Drulă, 1997).

Time Code-ul este o secvență de coduri numerice generate la un anumit interval de timp de către sistem. Sistemul de indexare *Time Code* este o metodă care folosește timpul pentru codificarea semnalului video. Metoda a fost perfectată de Society of Motion Picture and Television Engineers (SMPTE) ca un standard de sincronizare universal, utilizat în editarea video, audio și a filmului. Conform acestei codificări, fiecărui cadru video individual i se asociază câte un număr unic, ce permite controlul exact al benzii și indexarea secvențelor. Codificarea se poate realiza în diferite formate: time code *liniar* (LTC) sau *vertical* (VTC) (Wiki, 2009). Prin aceste numere se pot adresa cadrele individuale ale unei secvențe prin codul *timp* și apoi se pot accesa direct în timpul procesului de editare. Numărul unic al fiecărei imagini se codifică într-un bloc de 80 de biți, ce reprezintă în fapt, momentul captării sau înregistrării acesteia. Astfel, *Time Code* indică prin codajul BCD (Binary Coded Decimal) ora, minutul, secunda și numărul imaginii din fluxul video. Deoarece semnalul *Time Code* este înregistrat pe o pistă audio paralelă, codul s-a mai numit și *Liniar Time Code* sau *LTC*. Acest mod de a ține evidența cadrelor, poate pierde cu ușurință sincronismul imagine - numărul unic *Time Code*, datorită paralelismului semnalului cu banda video, la o simplă defazare a semnalului. Datorită acestui lucru, un alt format de codificare *Time Code* este cel vertical care apare ca o pereche de linii verticale negre și albe incluse în semnalul video. Aceste linii sunt de obicei inserate în intervalul neutilizat dintre două cadre consecutive ale unui flux video și exprimă același lucru ca și LTC.

Un alt sistem de reperaj și indexare a cadrelor video este *Frame Code*. Din punctul de vedere al codificării, el este mai simplu și se bazează pe numerotarea crescătoare a imaginilor fixe din fluxul video, începând cu primul cadru până la 300000.

În cadrul interfeței API sunt puse la dispoziția utilizatorilor funcții care permit conversia între cele două sisteme de reperaj (Dârdală et al., 2008). Astfel, funcția:

```
STDAPI_ (LONG) AVISampleToTime(PAVISTREAM p, LONG f);
```

(MSDN, 2009) convertește poziția cadrului f din fluxul p dată prin numărul său (*Frame Code*) în *Time Code*-ul corespunzător, în sensul că, funcția returnează timpul exprimat în milisecunde. Conversia, în sens invers, se realizează prin funcția:

STDAPI_ (LONG) AVIStreamTimeToSample(PAVISTREAM p, LONG t);

(MSDN, 2009) care primește poziția cadrului, din fluxul p , exprimată în milisecunde (t) și returnează poziția acestuia prin *Frame Code*-ul său.

Având disponibile aceste funcții de conversie putem face ca orice poziționare în flux să o exprimăm fie prin *Time Code* fie prin *Frame Code*.

3. Caracterul temporal și spațial al sincronizării interacțiunii într-un sistem hypervideo

Pentru implementarea unui sistem hypervideo trebuie să existe următoarele componente (Silver, 2009):

- o secvență video care să joace rol de element de legătură;
- un modul de sincronizare spațială și temporală; și
- un player de derulare a secvențelor video și care să gestioneze evenimentele de interacțiune cu utilizatorul.

Tehnicile de sincronizare în cadrul unui sistem hypervideo trebuie să satisfacă în primul rând caracterul temporal al realizării legăturii dintre entitățile informaționale. Cu alte cuvinte, în cadrul derulării unei secvențe video doar în anumite momente de timp se pot realiza legături cu alte entități informaționale. Există sisteme de tip hypervideo care limitează sincronizarea doar la dimensiunea temporală a datei video. În acest caz, este suficientă folosirea unei structuri care realizează indexarea secvenței video. Perioadele de timp când se pot efectua astfel de legături depind de obiectele vizualizate în fluxul video.

Spre exemplificare, se presupune existența unei date video de 12 cadre [1,13] pentru care se crează două legături pe baza a două entități:

- $o1$ ce ocupă secvența de cadre [4,8); și
- $o2$ care ocupă secvența de cadre [2,4) și [7,13].

În figura 1, se prezintă modul în care cele două entități cu semnificație de obiecte de legătură sunt repartizate în cadrele secvenței video. Această distribuție poartă numele de *harta asocierii* (Adah et al., 1996). Prin reuniunea intervalelor se obține perioada de timp senzitivă la evenimente de interacțiune cu utilizatorul, în exemplul prezentat intervalul este [2,13).

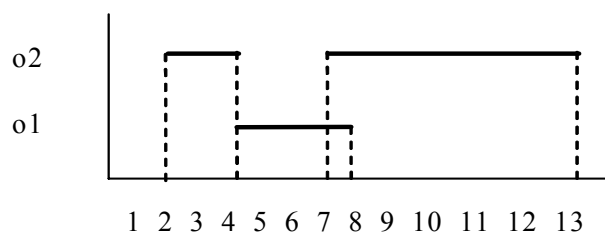


Figura 1. Acoperirea cadrelor de către obiectele o1 și o2

Astfel, player-ul video trebuie să fie sensibil la evenimentul de *click* pe fereastra de vizualizare pe toată perioada acoperită de obiectele din secvență, care joacă rol de ancore. În acest fel, sistemele hypervideo furizează elementele de legătură în mod dinamic pe măsura derulării secvenței.

Pe lângă caracterul temporal, în cadrul unui sistem hypervideo intervine și caracterul spațial al realizării legăturii (Chambel & Guimaraes, 2001). Spațialitatea se referă la faptul că, într-un cadru din secvență pot exista mai multe obiecte care fac trimiteri la diferite entități informaționale. În figura 2, se ilustrează un cadru ce are prezente ambele obiecte care realizează legături cu alte entități informaționale, prin declanșarea evenimentului *click*. În exemplul prezentat, se observă din harta asocierii, că în secvența video doar cadrul 7 are prezente ambele obiecte de legătură.

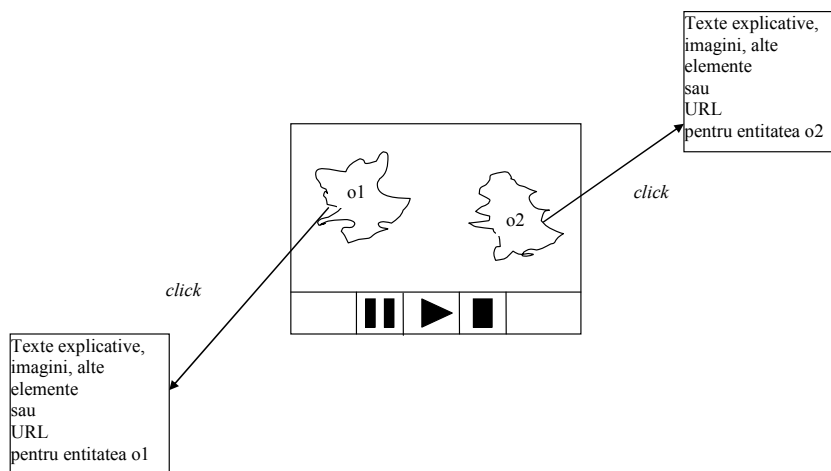


Figura 2. Sincronizare spațială

Devine evident că un astfel de sistem trebuie să gestioneze legăturile în mod sincron cu momentul derulării secvenței pe de o parte, iar pe de altă parte, să realizeze corect legătura în funcție de obiectul din cadrul video care a recepționat evenimentul.

O structură de date capabilă să gestioneze legăturile, luând în considerare atât caracterul temporal cât și spațial, în cadrul unui sistem hypervideo este arborele de segmentare cadre.

Arborele de segmentare cadre a fost gândit ca un instrument de realizare a indexării video. Adaptând informațiile conținute în această structură de date și definind noi operații, ea poate fi eficient folosită pentru a gestiona sincronizarea interacțiunii cu utilizatorul în cadrul unui sistem hypervideo.

4. Arborele de segmentare cadre adaptat pentru sistemele hypervideo

Un arbore de segmentare cadre (Adah et al., 1996) conține două tipuri de informații:

- *Primul tip* este dat de un set de entități care prezintă interes în film (caracteristicile) și care formează ancorele unui sistem hypervideo. În legătură cu entitățile, se păstrează și informații de localizare a lor în secvența video.
- *Al doilea tip* de informații se referă la cadrele video în care entitățile apar care constituie componenta temporală a unui sistem hypervideo.

O dată de tip video este compusă dintr-o succesiune de imagini (cadre) cu scopul de a prezenta cât mai fidel realitatea (în dinamică). O *secvență de cadre*, prin definiție este o pereche $[i, j)$ unde $1 \leq i \leq j \leq n$. i reprezintă primul cadru din film iar n este numărul ultimului cadru al filmului. $[i, j)$ reprezintă setul tuturor cadrelor între i (inclusiv) și j (exclusiv). Cu alte cuvinte, subsecvența $[i, j)$ este dată de mulțimea tuturor cadrelor k ce au proprietatea că se află între i și j :

$$[i, j) = \{ k / i \leq k < j \},$$

unde:

- i reprezintă cadrul de început;

- j cel de sfârșit al secvenței de cadre.

Se poate defini o relație de ordine (\subseteq) pe setul tuturor secvențelor de cadre, astfel: $[i_1, j_1] \subseteq [i_2, j_2]$ dacă $i_1 < j_1 \leq i_2 < j_2$. Intuitiv, $[i_1, j_1] \subseteq [i_2, j_2]$ înseamnă că secvența de cadre indicată de $[i_1, j_1]$ precede secvența de cadre indicată de $[i_2, j_2]$. De obicei, se utilizează $[i_1, j_1) \subset [i_2, j_2)$ unde $j_1 \neq i_2$.

Un set de secvențe de cadre X este *bine ordonat* dacă are următoarele proprietăți:

1. X este finit, adică $X = \{ [i_1, j_1), [i_2, j_2), \dots, [i_r, j_r) \}$ cu r dat;
2. $[i_1, j_1) \subseteq [i_2, j_2) \subseteq \dots \subseteq [i_r, j_r)$.

Setul X este *riguros constituit* dacă:

1. X este bine ordonat;
2. Nu există nici o secvență de cadre din X de forma $[i_1, i_2)$ și $[i_2, j_3)$.

Dacă ENT reprezintă mulțimea entităților unei secvențe video, o tabelă de asociere este o funcție λ astfel încât pentru fiecare $ent \in ENT$, $\lambda(ent)$ este un set de secvențe de cadre riguros constituit:

$$\lambda(ent) = \{ [i_1, j_1), [i_2, j_2) \},$$

ce presupune că în toate secvențele de cadre $[i_1, j_1)$ și $[i_2, j_2)$, entitatea ent este prezentă.

O hartă a asocierii este reprezentată în figura 1 și prezintă modul în care cele două entități se suprapun peste cadrele filmului.

Implementarea în aplicații a hărților de asociere presupune construirea unor structuri de date adecvate în acest sens. O astfel de structură este arborele de segmentare cadre.

Intr-un arbore de segmentare cadre, fiecare nod conține o secvență de cadre $[x,y)$ în care x reprezintă cadrul de început (inclusiv) iar y este cel final (exclusiv). Numărul fiecărui nod joacă rol de identificator al nodului iar setul de numere existent într-un nod reprezintă identificatorii entităților care apar în secvența de cadre asociată aceluia nod.

Informația utilă a nodului arborelui de segmentare cadre conține:

- limita minimă și maximă a intervalului;
- obiectele video conținute în nod; și

- informații de localizare a obiectelor în secvența indicată.

Pentru harta de asociere prezentată în figura 1, arborele este descris în figura 3.

Din figura 3 reies caracteristicile unui arbore de segmentare cadre:

- este un arbore binar;
- nodul rădăcină conține ca limită inferioară, respectiv superioară a intervalului numărul primului și al ultimului cadru plus 1, adică acoperă întregul film;
- se dezvoltă de la rădăcină către frunze pe principiul *divide and conquer* (prin înjumătățirea intervalului);
- dacă se asociază o entitate video unui nod, atunci intervalul memorat în nod constituie o secvență din setul de secvențe de cadre bine ordonat a entității respective.

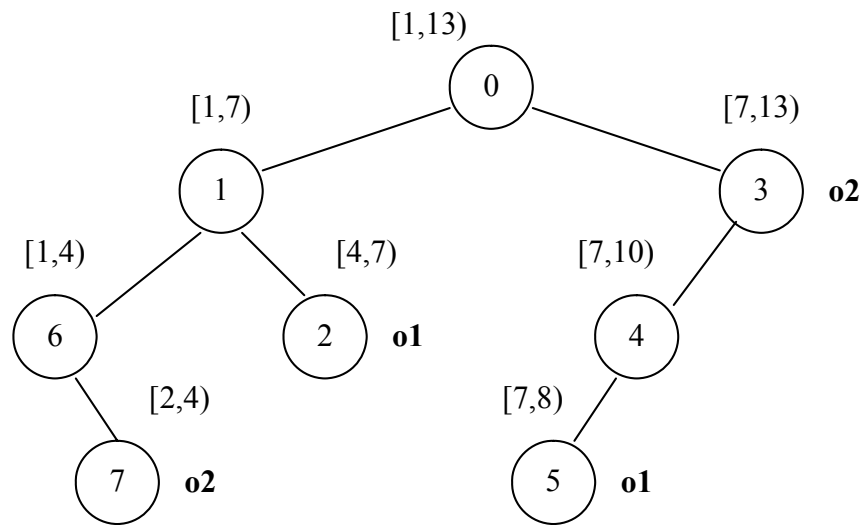


Figura 3. Arborele de segmentare cadre

Comportamentul structurii de date, în ansamblu, conține metode specifice de lucru cu structurile de date dinamice cum ar fi: inserarea unei secvențe de cadre, modificarea și ștergerea ei (Dârdală et al., 2008).

Inserarea unei entități video presupune furnizarea unui set de secvențe de cadre riguros construit. Metoda propriu-zisă de inserare va realiza inserarea

doar a unei secvențe din set urmând a se repeta acest lucru până la epuizarea secvențelor din setul de secvențe riguros construit.

Algoritmul de *inserare* primește secvența sub formă de interval (limita inferioară – *lio* și limita superioară – *lso*), numărul entității de inserat precum și informația de localizare spațială a obiectului ancoră. Se mai cunoaște pentru fiecare nod existent sau care urmează a fi construit, limitele: inferioară – *lin* și superioară – *lsn*, calculate prin metoda înjumătățirii intervalului. Algoritmul este recursiv și presupune repartizarea entității video acelor noduri din arbore astfel încât să se acopere exact intervalul secvenței.

Algoritmul începe prin a testa dacă limita inferioară a obiectului este mai mică decât limita superioară a obiectului. În caz afirmativ, algoritmul se continuă, altfel, se termină.

- Dacă rădăcina e vidă se construiește un nou nod pentru care limitele (inferioară și superioară) sunt preluate din parametri de intrare, respectiv limita inferioară și superioară a nodului;
- Dacă $[lin, lsn] \subseteq [lio, lso]$, atunci numărul entității se repartizează nodului, iar *lio* devine *lsn*;
- Se calculează jumătatea intervalului nodului: $lj = (lin + lsn) / 2$;
- Dacă $lso \leq lj$, înseamnă că entitatea video urmează să fie repartizată nodurilor subarborelui din stânga, $lsn = lj$, algoritmul se va relua, altfel dacă $lso \geq lj$, înseamnă că entitatea video urmează să fie repartizată nodurilor subarborelui drept, $lin = lj$, algoritmul se va relua, altfel înseamnă că entitatea video urmează să fie repartizată ambilor subarbori, deci algoritmul se va relua atât pentru subarboarele din stânga cât și pentru cel din dreapta.

Algoritmul de *ștergere* a unei entități presupune traversarea arborelui și ștergerea numărului entității din vectorul de entități a nodului ce conține acea entitate. Nu se șterg noduri fizic, deoarece inserări ulterioare ale altor entități ar putea avea ca efect recrearea unor noduri. Dacă se șterg toate entitățile memorate, arborele nu se va mai stoca pe suport extern.

Modificarea setului de secvențe ce corespunde unei entități are drept consecință redistribuirea entităților în funcție de noile limite ale intervalului secvenței.

Din prezentarea structurii arborescente de segmentare cadre se observă că entitățile video care furnizează legăturile sunt identificate prin numere. Este

util ca acestea să fie exprimate și manipulate prin intermediul unei date tip șir de caractere. Memorarea propriu-zisă a denumirii entităților cât și a altor informații auxiliare, ca de exemplu informația referită prin declanșarea evenimentului de interacțiune sau URL-ul, devine inefficientă dacă se face în fiecare nod al arborelui.

În aceste condiții, se propune construirea unei structuri de date suplimentare mai precis o listă ce va conține entitățile și referirile către nodurile din arborele de segmentare cadre, corespunzătoare setului de secvențe video. Figura 4 prezintă lista entităților video corespunzătoare arborelui de segmentare cadre din figura 3. Din figura 4 se observă că obiectul o1 este distribuit în secvența video în intervalele de cadre existente în nodurile 2 și 5 din arborele de segmentare cadre (figura 3), adică [4,7) și [7,8), prin reuniunea intervalelor rezultă intervalul [4,8), în timp ce obiectul o2 este distribuit în intervalele existente în nodurile 7 și 3 din arbore, adică intervalele [2,4) și [7,13).

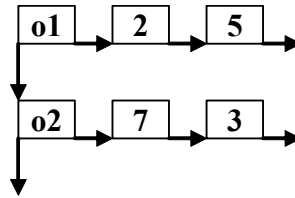


Figura 4. Lista entităților video

O aplicație care dezvoltă un sistem hypervideo exploatează această structură pentru a prelua informațiile de sincronizare. Astfel, când *player*-ul interceptează evenimentul *click* se parcurg următoarele etape în procesare:

- se interoghează numărul cadrului curent vizualizat;
- se exploatează structura arborescentă și se verifică dacă la acel moment sunt prezente obiecte care definesc legături; dacă nu procesul se termină, altfel se continuă cu următoarele etape;
- se preiau coordonatele cursorului mouse-ului și pentru fiecare obiect ancoră prezent în cadrul curent se efectuează operația de intersecție dintre punctul având coordonatele cursorului mouse-ului și regiunea de localizare a obiectului; dacă intersecția este vidă, pentru toate obiectele, atunci procesul se termină, altfel se identifică obiectul ancoră și algoritmul continuă;

- se preia informația de legătură din lista entităților video și se afișează sau se face trimitere la URL-ul specificat în structură și se oprește temporal derularea secvenței video.

5. Concluzii

Utilitatea dezvoltării de tehnici de construire a sistemelor hypermedia este cu atât mai importantă cu cât prezentările video încearcă să ocupe un rol dominant în aplicațiile multimedia. Acestea se regăsesc tot mai mult în aplicații multimedia de prezentare, instruire și divertisment. Structurile de date eficiente sunt folosite pentru a gestiona informații de sincronizare, indexare și adnotare iar acestea pot fi ușor reprezentate prin standarde de descriere a datelor video cum ar fi MPEG7 și MPEG21.

Referințe

- Adah, S., Selcuk Candan, K., Chen, S., Erol, K., Subrahmanian, V.S., *The Advanced Video Information System: data structures and query processing*. Multimedia Systems, vol 4, Springer Verlag, 1996;
- Chambel, T., Guimaraes, N., *Learning with Video in Hypermedia*. Technical Reports, Department of Informatics, University of Lisbon, 2001, <http://hdl.handle.net/10455/3017>;
- Dârdală, M., Reveiu, A., *Techniques for Media Stream Synchronization*, The Proceedings of the Ninth International Conference of Informatics in Economy, ASE București, 2009, în volumul Education, Research & Business Technologies, ASE Printing House, București, 2009;
- Dârdală, M., Reveiu, A., Furtună, T. F., *Sincronizarea secvențelor video cu fluxuri de tip text*, Volumul de lucrări ale celei de a șasea Conferințe Naționale de Interacțiune Om-Calculator, RoCHI 2009, Universitatea Tehnică din Cluj-Napoca, Cluj-Napoca, 3-4 Septembrie 2009, Editura MatrixRom, București, 2009;
- Dârdală, M., Smeureanu, I., Reveiu, A., *Tehnologii multimedia*. Editura ASE, București, 2008;
- Deb, S., *Video Data Management and Information Retrieval*, IRM Press, 2005;
- Paganini, M., *Encyclopedia of Multimedia Technology and Networking*, Idea Group Inc., 2005;
- Reveiu, A., Dârdală, M., Furtuna, F., *Using Content-based Multimedia Data Retrieval for Multimedia Content Adaptation*, Proceedings of the 12th Int. Conf. on Human-Computer Interaction HCI'2007 (Beijing, 22-27 July 2007), Vol. 3, LNCS 4552: Lecture Notes in Computer Science, Springer-Verlag 2007;

Smeureanu, I., Drulă, G., *Multimedia concepte și practică*, Ed. CISON, București, 1997;
***, Microsoft Press - MSDN, 2009;
<http://en.wikipedia.org/wiki/Hypervideo>, 2009
<http://silverlight.net/learn/tutorials/hypervideo-cs/>, 2009