

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ  
ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΕΠΙΣΤΗΜΗ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εργασία για το μάθημα:  
Διαχείριση Σύνθετων Αντικειμένων Πολλαπλών Μέσων σε Β. Δ.

**Θέματα Συγχρονισμού σε Κατανεμημένες Πολυμεσικές  
Παρουσιάσεις**

Όνομα:  
ΤΡΑΧΑΝΗ ΒΑΣΙΛΙΚΗ

ΑΜ:  
925

Μάρτιος 2014

## **ΠΕΡΙΕΧΟΜΕΝΑ**

<b>1. ΕΙΣΑΓΩΓΗ.....</b>	<b>4</b>
1.1 ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ.....	4
1.2 ΠΕΡΙΕΧΟΜΕΝΑ.....	5
<b>2. ΤΕΧΝΟΛΟΓΙΚΑ ΘΕΜΑΤΑ.....</b>	<b>6</b>
2.1 ΣΧΕΤΙΚΕΣ ΤΕΧΝΟΛΟΓΙΕΣ.....	6
2.2 ΠΟΛΥΜΕΣΙΚΕΣ ΠΑΡΟΥΣΙΑΣΕΙΣ ΚΑΙ ΣΥΓΧΡΟΝΙΣΜΟΣ.....	8
2.3 ΠΡΟΤΥΠΟ SMIL.....	13
<b>3. ΣΧΕΤΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>15</b>
3.1 ΠΑΡΟΥΣΙΑΣΗ.....	15
3.2 ΚΡΙΤΙΚΗ.....	19
<b>4. ΥΛΟΠΟΙΗΣΗ ΜΟΝΤΕΛΟΥ.....</b>	<b>25</b>
4.1 ΘΕΩΡΗΤΙΚΗ ΕΠΕΞΗΓΗΣΗ ΜΟΝΤΕΛΟΥ.....	26
4.2 BROADCASTER.....	29
4.3 VIEWER.....	34
4.4 ΠΕΙΡΑΜΑΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ.....	35
<b>5. ΣΥΜΠΕΡΑΣΜΑΤΑ.....</b>	<b>42</b>
<b>ΑΝΑΦΟΡΕΣ.....</b>	<b>44</b>
<b>APPENDIX.....</b>	<b>45</b>

## Σύνοψη

Ένα από τα δισεπίλυτα προβλήματα του World Wide Web είναι η ενσωμάτωση και σωστή διαχείριση πολυμεσικών αρχείων που περιέχουν είτε βίντεο είτε ήχο. Τα πρότυπα που κυριαρχούν αυτή τη στιγμή και τα αντίστοιχα πρωτόκολλα δεν μπορούν να εξασφαλίσουν την ομαλή χρονική μετάδοση πολυμεσικών παρουσιάσεων στο διαδίκτυο και παρουσιάζουν προβλήματα συγχρονισμού. Το μεγαλύτερο πρόβλημα όσον αφορά το συγχρονισμό πολυμεσικών χωρο-χρονικών εφαρμογών παραμένει ότι χρειάζεται να γνωρίζουμε εκ των προτέρων πότε θα συμβεί ένα γεγονός μέσα στην παρουσίαση για να επιτευχθεί ο σωστός συγχρονισμός.

Στην παρούσα αναφορά θα αναφερθούμε κυρίως σε διαδραστικές πολυμεσικές παρουσιάσεις που μεταδίδονται ζωντανά, όπως ένας ποδοσφαιρικός αγώνας ή η απονομή των βραβείων Oscar. Θα εξετάσουμε τις τρέχουσες τεχνολογίες, παρουσιάζοντας τις λύσεις που προσφέρουν αλλά και τα κύρια προβλήματά τους, δίνοντας έμφαση στην markup γλώσσα SMIL.

Κατόπιν, θα παρουσιάσουμε ένα νέο ολοκληρωμένο σύστημα δημιουργίας και διαχείρισης SMIL αρχείων σε μια ζωντανή, διαδικτυακή, πολυμεσική παρουσίαση. Το μοντέλο που θα προτείνουμε θα υποστηρίζει τον συγχρονισμό γεγονότων που ξέρουμε εκ των προτέρων ότι θα συμβούν μέσα στην παρουσίαση αλλά που δεν είναι ορισμένα σε συγκεκριμένο χρόνο. Παράλληλα, το σύστημα θα προσφέρει λύση για το πρόβλημα του Late tuning, δηλαδή να μπορεί ένας χρήστης να παρακολουθήσει ετεροχρονισμένα κάποια γεγονότα που συνέβησαν αλλά τα έχασε λόγω ότι δεν συνδέθηκε από την έναρξη της ζωντανής μετάδοσης.

# 1. ΕΙΣΑΓΩΓΗ

Οι πολυμεσικές (multimedia) εφαρμογές, όπως για παράδειγμα ένα παιχνίδι στον υπολογιστή, παρουσιάζουν μια δυνατή και δημοφιλή μέθοδο επικοινωνίας που ενσωματώνει ποικιλία medias με διαφορετικούς χρονικούς περιορισμούς. Λόγω της σύγχυσης που υπάρχει στον όρο πολυμέσα στην παρούσα αναφορά όταν αναφερόμαστε σε πολυμεσικές παρουσιάσεις θα εννοούμε παρουσιάσεις που περιέχουν τόσο αντικείμενα που δεν έχουν σχέση με το χρόνο, όπως για παράδειγμα εικόνες και κείμενο, όσο και αντικείμενα που έχουν τη βάση τους στο χρόνο, όπως είναι το βίντεο, ο ήχος και το animation.

Πιο συγκεκριμένα, στην παρούσα αναφορά, θα μελετήσουμε διαδραστικές πολυμεσικές παρουσιάσεις που περιέχουν σίγουρα ήχο και βίντεο. Ειδικότερα, θα εστιάσουμε σε ζωντανές παρουσιάσεις που μεταδίδονται είτε μέσω ψηφιακής τηλεόρασης είτε μέσα από το διαδίκτυο. Παραδείγματα τέτοιων παρουσιάσεων είναι ένα ζωντανό αθλητικό γεγονός ή μια μουσική συναυλία που μεταδίδεται απευθείας.

Με τον όρο διαδραστικές εννοούμε ότι ο δημιουργός αλλά και ο απλός θεατής της παρουσίασης θα μπορούν να αλληλεπιδρούν και να παρεμβαίνουν στην προκαθορισμένη ροή της παρουσίασης. Η ενεργή συμμετοχή του δημιουργού ή του θεατή μπορεί να μετατρέψει μια παρουσίαση από βαρετή, μονοδιάστατη και συντηρητική σε μια ενδιαφέρουσα και αλληλεπιδραστική παρουσίαση που θα ταιριάζει στο προφίλ του χρήστη.

Η αλληλεπίδραση των συμμετεχόντων σε μια τέτοιου είδους παρουσίαση γίνεται μέσω γεγονότων που ξέρουμε ότι μπορεί να συμβούν αλλά δεν γνωρίζουμε εκ των προτέρων πότε πρόκειται να γίνει αυτό. Αυτό δημιουργεί αυτόματα πρόβλημα συγχρονισμού αφού διαφορετικά βίντεο πρέπει να συνδυασθούν να εκτελούνται παράλληλα είτε σε ακολουθία με χρονικές σχέσεις που πρέπει να ορισθούν δυναμικά. Παράλληλα, θα μελετήσουμε το πρόβλημα του late tuning, δηλαδή της δυνατότητας που πρέπει να δίνεται στον θεατή που δεν παρακολούθησε από την αρχή τη μετάδοση μιας παρουσίασης να παρακολουθεί ετεροχρονισμένα τα σημαντικότερα γεγονότα που έχασε.

Στην επόμενη ενότητα, θα αναλύσουμε το γενικότερο πρόβλημα του συγχρονισμού αλλά και ειδικότερα το πρόβλημα του late tuning πριν προχωρήσουμε να περιγράψουμε στα επόμενα κεφάλαια το μοντέλο που προτείνουμε για τη λύση του.

## 1.1 ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Παρά την ραγδαία διάδοση του διαδικτύου και τις πολλές σύγχρονες εφαρμογές που έχουν αναπτυχθεί, η διαχείριση των πολυμεσικών παρουσιάσεων αλλά και των αρχείων που περιέχουν βίντεο και ήχο μέσα στο διαδίκτυο παραμένει προβληματική. Το πρωτόκολλο HTTP αποδείχθηκε

ανεπαρκές σε αυτόν τον τομέα [3] αλλά και οι νέες προσπάθειες με τις τεχνολογίες HTML5 και XHTML, παρότι είναι εξαιρετικά φιλόδοξες παραμένουν ανεπαρκείς.

Στην παρούσα αναφορά δεν θα ασχοληθούμε καθόλου με τις εφαρμογές που δεν περιέχουν ήχο ή βίντεο, όπως οι εφαρμογές e-mail και TV-Banking. Θα αναφερθούμε, λοιπόν, μόνο σε πολυμεσικές εφαρμογές που περιέχουν βίντεο και ήχο και στο εξής θα αποκαλούμε και shows. Ιδιαίτερα, θα μελετήσουμε τις περιπτώσεις ζωντανών πολυμεσικών παρουσιάσεων που μεταδίδονται είτε μέσω της ψηφιακής τηλεόρασης IDTV (Interactive Digital TV) είτε μέσα από το διαδίκτυο.

Σε αυτές, λοιπόν, τις περιπτώσεις το πρόβλημα του συγχρονισμού εστιάζεται περισσότερο στο γεγονός πως δεν γνωρίζουμε εκ των προτέρων το περιεχόμενο τους. Σε μια στατική πολυμεσική παρουσίαση ο δημιουργός της ξέρει τι περιέχει από πριν με αποτέλεσμα να μπορεί να συγχρονίσει κατάλληλα τα πολυμεσικά αντικείμενα. Έτσι, για παράδειγμα, μπορεί να συντονίσει μια διαφήμιση ενός χορηγού με ένα συγκεκριμένο χρονικό σημείο του βίντεο. Αντίθετα, στις ζωντανές μεταδόσεις το περιεχόμενο δεν είναι γνωστό εκ των προτέρων. Γνωρίζουμε ότι κάποια γεγονότα μπορεί ενδεχομένως να συμβούν, όπως η επίτευξη ενός γκολ σε έναν ποδοσφαιρικό αγώνα, αλλά δεν γνωρίζουμε πόσες φορές θα συμβεί το γεγονός αλλά και πότε χρονικά θα γίνει αυτό μέσα στην παρουσίαση.

Επίσης, συμβαίνει πολλές φορές ένας θεατής να μην παρακολουθήσει μια ζωντανή μετάδοση από την αρχή και να χάσει κάποια σημαντικά γεγονότα. Χαρακτηριστικό παράδειγμα είναι η εκκίνηση σε έναν αγώνα φόρμουλα1 ή η επίτευξη ενός γκολ στην αρχή ενός ποδοσφαιρικού αγώνα, γεγονότα που από εδώ και στο εξής θα αποκαλούμε και ως highlight. Ο θεατής θα πρέπει να έχει την δυνατότητα παράλληλα με την ζωντανή μετάδοση να δει και όσα highlight έχει χάσει, σενάριο που από εδώ και πέρα θα αποκαλούμε ως Late tuning.

Το μοντέλο που θα παρουσιάσουμε φιλοδοξεί να δώσει λύση στα προβλήματα συγχρονισμού σε ζωντανές πολυμεσικές παρουσιάσεις αλλά και στο πρόβλημα του Late tuning, παρέχοντας ένα ολοκληρωμένο σύστημα που θα δίνει έγκυρα και έγκαιρα στο χρήστη την δυνατότητα να προβάλλει τα γεγονότα που έχασε.

## **1.2 ΠΕΡΙΕΧΟΜΕΝΑ**

Στην παρούσα ενότητα θα περιγράψουμε συνοπτικά τη δομή της παρούσας αναφοράς. Στο επόμενο κεφάλαιο αναφερόμαστε στα σύγχρονα εμπορικά συστήματα αλλά και στις τεχνολογίες που αφορούν την εργασία μας με κυριότερη έμφαση στο πρότυπο SMIL. Στο τρίτο κεφάλαιο παρουσιάζουμε τη σχετική βιβλιογραφία γύρω από το θέμα μας, μελετώντας τα πλεονεκτήματα και μειονεκτήματά της κάθε περίπτωσης και ελέγχοντας πόσο μπορεί να συνεισφέρουν στην υλοποίησή του μοντέλου μας. Στο τέταρτο κεφάλαιο αναφερόμαστε στο θεωρητικό μοντέλο του συστήματος που προτείνουμε και δείχνουμε αναλυτικά την υλοποίησή του αλλά και την πειραματική αξιολόγησή του. Τέλος, στο

πέμπτο κεφάλαιο καταδεικνύουμε τα κύρια συμπεράσματα της αναφοράς μας και προτείνουμε μελλοντικές ιδέες που μπορούν να υλοποιηθούν για να βελτιώσουν το σύστημά μας αλλά και ερωτήματα που μένουν ανοικτά.

## **2. ΤΕΧΝΟΛΟΓΙΚΑ ΘΕΜΑΤΑ**

Στο κεφάλαιο αυτό θα μελετήσουμε την υπάρχουσα τεχνολογία και τα επικρατέστερα σύγχρονα εμπορικά συστήματα για την δημιουργία πολυμεσικών εφαρμογών στο διαδίκτυο. Ακολούθως, θα αναφερθούμε στις διαφορετικές περιπτώσεις συγχρονισμού που πρόκειται να εξετάσουμε με αναφορά σε συγκεκριμένα παραδείγματα και τέλος θα περιγράψουμε το πρότυπο SMIL που πρόκειται να χρησιμοποιήσουμε στην υλοποίηση του συστήματός μας.

### **2.1 ΣΧΕΤΙΚΕΣ ΤΕΧΝΟΛΟΓΙΕΣ**

Οι μέθοδοι αντιμετώπισης του προβλήματος της διαχείρισης των πολυμεσικών παρουσιάσεων στο διαδίκτυο μπορούν να διαχωριστούν σε διαδικαστικές-προστακτικές (procedural - imperative) και δομημένες – δηλωτικές (structured – declarative). Από τις δύο αυτές εναλλακτικές, αυτή που έχει επικρατήσει στη σύγχρονη εποχή είναι η διαδικαστική προσέγγιση με βάση κάποιες γλώσσες προγραμματισμού ενώ αντίθετα η δηλωτική προσέγγιση που έχει σαν βάση κάποιο πρότυπο, παρότι κερδίζει συνεχώς πόντους, δεν έχει καταφέρει να γίνει ακόμη δημοφιλής.

Αυτή τη στιγμή οι επικρατέστερες τεχνολογίες με βάση την διαδικαστική μέθοδο είναι η τεχνολογία Flash [13] και η γλώσσα προγραμματισμού JavaScript. Η χρησιμοποίηση του Flash είναι η πλέον διαδεδομένη και χρησιμοποιείται σε πληθώρα εφαρμογών με κυριότερο παράδειγμα το YouTube [4]. Η διαχείριση αλλά και η αλληλεπίδραση με το χρήστη γίνεται αποκλειστικά μέσα από την υλοποίηση σε Action Script που σημαίνει ότι ο δημιουργός της πολυμεσικής παρουσίασης πρέπει να έχει ειδικές τεχνολογικές γνώσεις. Παράλληλα, λόγω της δυαδικής μορφής της εφαρμογής Flash, το περιεχόμενο δεν είναι εύκολα ανακτήσιμο και προσβάσιμο από κάποια άλλη εξωτερική εφαρμογή ή από τον απλό χρήστη του διαδικτύου.

Η άλλη επιλογή της JavaScript μπορεί και αυτή να συντελέσει στην δημιουργία πολυμεσικών παρουσιάσεων με τη βοήθεια της HTML αλλά και σε συνδυασμό με ένα μέσο αναπαραγωγής πολυμεσικών αρχείων, όπως είναι το Ambulant Player. Είναι γενικά πιο βατή στην υλοποίηση από την τεχνολογία Flash και μπορεί να παρέχει ορισμένες δυνατότητες αλληλεπίδρασης με τον χρήστη. Πιο συγκεκριμένα, μπορεί κανείς να δώσει δυνατότητες στον χρήστη για την εκκίνηση ή διακοπή της αναπαραγωγής ενός πολυμέσου αλλά δεν μπορεί να του προσφέρει αλληλεπίδραση βασισμένη σε γεγονότα ή σε συνθήκες που έχουν ως κύριο χαρακτηριστικό το χρόνο. Έτσι, συνήθως, παρουσιάζεται

ως λιγότερη αποτελεσματική από την τεχνολογία Flash ιδιαίτερα σε διαδραστικές, δυναμικές πολυμεσικές παρουσιάσεις.

Από την άλλη πλευρά, μια αξιέπαινη προσπάθεια που ξεφεύγει από τη λογική των γλωσσών προγραμματισμού και θέλει να χρησιμοποιήσει μια πιο δομημένη λογική είναι η τεχνολογία XForms. Η τεχνολογία XForms χρησιμοποιεί τη μορφοποίηση XML για να ορίσει ένα μοντέλο δεδομένων και να δηλώσει όχι μόνο τις διαδικτυακές φόρμες αλλά και την σχέση αυτών των φορμών με τα δεδομένα, προσπαθώντας να προωθήσει την διαδραστικότητα και την εύκολη πρόσβαση στο χρήστη [6]. Η έκδοση XForms 1.1 δεν διαθέτει το δικό της μοντέλο υλοποίησης παρά μόνο ένα απλό σύστημα ορισμού παραμέτρων, σαν ένα φύλλο excel, που μπορούν να υπολογιστούν σε συνάρτηση άλλων παραμέτρων.

Στην ίδια λογική το πρότυπο SMIL (Synchronized Multimedia Integration Language) ξεκίνησε με μεγάλες προσδοκίες πως θα λύσει το πρόβλημα του συγχρονισμού και της αλληλεπίδρασης σε πολυμεσικές παρουσιάσεις, παρουσιάζοντας μια δομημένη γλώσσα που είχε ως κύριο συστατικό τον χρόνο. Εντούτοις, σήμερα δεν θεωρείται ανταγωνιστής των τεχνολογιών Flash και JavaScript στον τομέα των πολυμεσικών παρουσιάσεων. Επειδή, η γλώσσα SMIL θα αποτελέσει σημαντικό κομμάτι του μοντέλου μας θα την παρουσιάσουμε αργότερα σε επόμενη ενότητα και θα περιοριστούμε να αναφέρουμε προς το παρόν τις εμπορικές εφαρμογές που υποστηρίζουν την SMIL.

Οι πιο διαδεδομένες εφαρμογές αναπαραγωγής πολυμέσων με την υποστήριξη της γλώσσας SMIL 2.0 είναι αυτή τη στιγμή η εφαρμογή RealPlayer, το πρόγραμμα Internet Explorer έκδοση 6 και η εφαρμογή Ambulant SMIL Player. Στη συνέχεια θα αναλύσουμε τα πλεονεκτήματα και τα μειονεκτήματα της κάθε μιας.

Η πρώτη εφαρμογή αναπαραγωγής αρχείων σε γλώσσα SMIL ήταν ο RealPlayer και κατόπιν ο διάδοχός του, η εφαρμογή RealOne. Πρόκειται για ανεξάρτητη stand alone εφαρμογή διαθέσιμη για τα περισσότερα λειτουργικά συστήματα που όμως έχει δεχτεί πολλές αρνητικές κριτικές από χρήστες [2] αλλά υποστηρίζει τη γλώσσα SMIL μέχρι την έκδοση 2.1.

Από την άλλη πλευρά, παρόλο που ο Internet Explorer 6 ήταν η πιο διαδεδομένη δικτυακή εφαρμογή για αρκετό καιρό, η υποστήριξή του όσον αφορά τη γλώσσα SMIL σταματούσε στα βασικά της σημεία και δεν υποστήριζε τα σύνθετα στοιχεία της όπως τη χρονική συνδεσμολογία σε μια παρουσίαση και ο αντίστοιχος συγχρονισμός. Από την έκδοση 7 και μετέπειτα η υποστήριξη της SMIL αντικαταστάθηκε από το πρότυπο SAMI (Synchronized Accessible Media Interchange) που αναπτύχθηκε αποκλειστικά για τα προϊόντα της Microsoft, οπότε η εφαρμογή του Internet Explorer δεν πρόκειται να μας απασχολήσει περαιτέρω.

Τέλος, η εφαρμογή του Ambulant SMIL Player είναι η μόνη που υποστηρίζει το πρότυπο της SMIL ακόμα και στην τελευταία έκδοση 3.0 [15]. Διατίθεται δωρεάν είτε ως ανεξάρτητη εφαρμογή είτε ως

plug-in για αρκετούς browsers. Εκτός ότι παρουσιάζει το μειονέκτημα ότι δεν υποστηρίζει όλους τους browsers και όχι όλες τις εκδόσεις των υπολοίπων, διάφορα τεστ έχουν δείξει ότι μόλις η πολυπλοκότητα του συγχρονισμού αυξάνει σε μια πολυμεσική παρουσίαση, η εφαρμογή παρουσιάζει προβλήματα [2].

Πρέπει να αναφέρουμε ότι έχουν γίνει και προσπάθειες δημιουργίας νέων browser που θα διαχειρίζονται αποτελεσματικά τα SMIL αρχεία για την δημιουργία διαδραστικών πολυμεσικών παρουσιάσεων. Το SmilingWeb browser είναι μια τέτοια πλατφόρμα αναπαραγωγής πολυμέσων που υλοποιήθηκε με βάση τη JavaScript και αναπτύχθηκε με σκοπό την αναπαραγωγή παρουσιάσεων με την βοήθεια SMIL αρχείων μέσα από διαδικτυακές σελίδες. Η πλατφόρμα αυτή δημιουργήθηκε με τη βοήθεια βιβλιοθηκών της JavaScript έτσι ώστε να είναι ανεξάρτητη από το browser που χρησιμοποιεί ο χρήστης στο σπίτι του. Στην πλατφόρμα έγιναν δοκιμές και από το SMIL Testsuite της W3C με ικανοποιητικά αποτελέσματα [2]. Παρόλα αυτά, το SmilingWeb δεν υποστηρίζει όλες τις δυνατότητες της γλώσσας SMIL που παρουσιάστηκαν στην 3<sup>η</sup> έκδοση με αποτέλεσμα ο δημιουργός του αρχείου SMIL να μην είναι σίγουρος για το τελικό αποτέλεσμα. Επιπλέον, σε περιπτώσεις που το εύρος ζώνης δεν είναι ικανοποιητικό παρουσιάζονται σημαντικά προβλήματα συγχρονισμού κατά την διάρκεια μιας διαδικτυακής παρουσίασης. Ειδικά σε πολυμεσικές παρουσιάσεις με βίντεο και ήχο, θα παρουσιάζονταν στον χρήστη προβλήματα στην χρονική συμπεριφορά της παρουσίασης με τον χρήστη να παρατηρεί πολλά προβλήματα συγχρονισμού και παύσεις.

Τέλος, αναφέρουμε ότι υπάρχουν και πολλές ανεξάρτητες εφαρμογές που έχουν υλοποιηθεί σε γλώσσα JavaScript με σκοπό να γίνεται σωστή διαχείριση των SMIL αρχείων. Χαρακτηριστικό παράδειγμα είναι η πλατφόρμα smilText που μπορεί να κάνει τα αρχεία SMIL αναγνώσιμα στους περισσότερους σύγχρονους browsers.

## **2.2 ΠΟΛΥΜΕΣΙΚΕΣ ΠΑΡΟΥΣΙΑΣΕΙΣ ΚΑΙ ΣΥΓΧΡΟΝΙΣΜΟΣ**

Η συμπεριφορά των πολυμεσικών παρουσιάσεων όσον αφορά τα θέματα συγχρονισμού ποικίλει ανάλογα με τα πολυμέσα από τα οποία αποτελείται η κάθε παρουσίαση. Έτσι, ο συγχρονισμός στη περίπτωση μετάδοσης ενός βίντεο που διαθέτει τον δικό του ήχο είναι μια απλή υπόθεση. Αντίθετα, όταν μια παρουσίαση περιλαμβάνει πολλά διαφορετικά αρχεία βίντεο και ήχου που πρέπει να συγχρονιστούν τόσο σε χρόνο όσο και χωρικά, η περίπτωση δυσκολεύει σε μεγάλο βαθμό.

Στην παρούσα αναφορά θα ασχοληθούμε κυρίως με ζωντανές πολυμεσικές παρουσιάσεις που συνδυάζουν διαφορετικά πολυμεσικά αντικείμενα και ο συγχρονισμός τους είναι εξαιρετικά δύσκολος. Τέτοιες παρουσιάσεις που μεταδίδονται ζωντανά στο διαδίκτυο μέσω Live streaming ή μέσω της



ψηφιακής τηλεόρασης IDTV μπορούν να χωριστούν σε 3 κατηγορίες ανάλογα με το περιεχόμενό τους αλλά και τον βαθμό δυσκολίας σε θέματα συγχρονισμού [10]:

1. Το περιεχόμενο των παρουσιάσεων είναι ανεξάρτητο από άλλα πολυμεσικά αντικείμενα. Τέτοια χαρακτηριστική περίπτωση είναι η εφαρμογή Television Banking όπου ο θεατής μπορεί να διαχειριστεί τον τραπεζικό του λογαριασμό ανεξάρτητα από το τι παίζει η τηλεόραση εκείνη την στιγμή.
2. Το περιεχόμενο των παρουσιάσεων έχει σημασιολογική (semantic) σχέση με άλλα πολυμεσικά αντικείμενα αλλά δεν υπάρχουν αυστηροί περιορισμοί στα θέματα συγχρονισμού. Ένα τέτοιο σενάριο εμφανίζεται με την παρουσίαση των τιμών των μετοχών ενός χρηματιστηρίου κατά την διάρκεια μιας τηλεοπτικής εκπομπής οικονομικού περιεχομένου. Η σημασιολογική σχέση μεταξύ τους είναι εμφανής αλλά, όπως συνήθως γίνεται σε αυτές τις περιπτώσεις, η μετάδοση των τιμών γίνεται ανεξάρτητα από οποιοδήποτε άλλο βίντεο μεταδίδεται εκείνη τη στιγμή. Έτσι και σε αυτό το σενάριο δεν παρουσιάζονται προβλήματα συγχρονισμού.
3. Το περιεχόμενο αυτών των παρουσιάσεων είναι όχι μόνο σημασιολογικά σχετικό με άλλα πολυμέσα αλλά πρέπει και να μεταδοθεί συγχρονισμένα με βάση αυτά τα άλλα πολυμέσα. Ένα τέτοιο σενάριο αποτελεί η μετάδοση μιας διαφήμισης ενός προϊόντος σε συγκεκριμένη χρονική στιγμή μιας ταινίας.

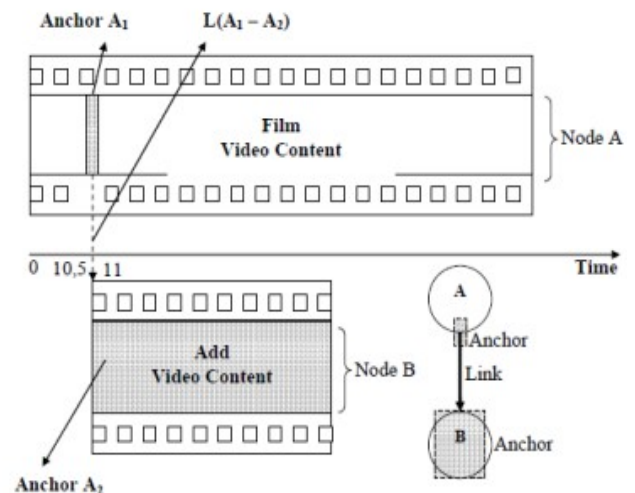
Οι δύο πρώτες περιπτώσεις δεν θα μας απασχολήσουν στην παρούσα αναφορά. Αντίθετα, θα ασχοληθούμε αποκλειστικά με την τελευταία περίπτωση που θέτει και τα πιο σοβαρά θέματα συγχρονισμού. Για αυτό το λόγο θα αναλύσουμε την τελευταία περίπτωση ανάλογα με το αν το περιεχόμενο της παρουσίασης είναι γνωστό εκ των προτέρων. Στην συνέχεια θα περιγράψουμε το σενάριο κατά το οποίο το περιεχόμενο είναι γνωστό εκ των προτέρων (content in advance) αλλά και τι σενάριο κατά το οποίο το περιεχόμενο δημιουργείται κατά την ζωντανή μετάδοση και δεν είναι γνωστό εξ αρχής (live generated content).

### **Content known in advance**

Στην περίπτωση αυτή γνωρίζουμε εκ των προτέρων το περιεχόμενο μιας πολυμεσικής παρουσίασης, αλλά δεν έχει οριστεί το ακριβές σημείο μετάδοσης. Ο συγχρονισμός επιτυγχάνεται μέσω της δημιουργίας anchors και διαδραστικών σημείων μέσα στην παρουσίαση. Τέτοιο χαρακτηριστικό σενάριο αποτελεί η παράλληλη εμφάνιση μιας διαφήμισης με συγκεκριμένο περιεχόμενο κατά την διάρκεια μιας ταινίας. Η διαφήμιση έχει δημιουργηθεί πριν από την ταινία, άρα το περιεχόμενο της είναι γνωστό, αλλά δεν γνωρίζουμε εκ των προτέρων πότε θα μεταδοθεί.

Για παράδειγμα θέλουμε να μεταδώσουμε μια διαφήμιση για γυαλιά ηλίου τη στιγμή που στην ταινία ένα ηθοποιός βάζει τα συγκεκριμένα γυαλιά ηλίου [8]. Η διαφήμιση έχει δημιουργηθεί πριν από την έναρξη της ταινίας και το περιεχόμενό της είναι γνωστό. Αυτό που απομένει είναι να ορίσουμε τα χρονικά όρια (anchors) στην ταινία κατά τα οποία μπορεί να εμφανιστεί η διαφήμιση για να λύσουμε το θέμα συγχρονισμού.

Όπως φαίνεται και στο σχετικό σχεδιάγραμμα, ο κόμβος A (Node A) αντιπροσωπεύει την ταινία και ο κόμβος B (Node B) αντιπροσωπεύει τη διαφήμιση. Ορίζουμε το anchor A1 που αναφέρεται σε ένα συγκεκριμένο μέρος της ταινίας, μεταξύ του 10.5<sup>ου</sup> και 11<sup>ου</sup> λεπτού και το anchor A2 που αναφέρεται σε όλο το βίντεο της διαφήμισης. Και τα δύο anchors είναι γνωστά εκ των προτέρων και μπορούν να ορισθούν πριν ξεκινήσει η μετάδοση της παρουσίασης. Ο ορισμός γίνεται μέσα σε ένα αρχείο SMIL όπως φαίνεται και παρακάτω. Ο



συγχρονισμός ορίζεται από τη δημιουργία και εισαγωγή της σύνδεσης link L(A1-A2) που θα πυροδοτήσει την έναρξη της διαφήμισης στο συγκεκριμένο χρονικό διάστημα.

```
<video id = "movie" region="show_video"
  src = "rstp://www.movies.com/MIB"
  dur = "smpte=02:12:19:00" >
  <anchor id = "A1"
    begin = "smpte=00:10:30:00"
    end = "smpte=00:11:00:00"
    coords = "0%, 0%, 50%, 50%"
    href = "http://www.advertisement.com/sunglasses"/>
  <anchor id = "A2" begin = "3555s" end = "3600s"
    coords = "50%, 50%, 50%, 50%"
    href = "http://www.advertisement.com/optics"/>
</video>
```

SMIL (Content in advance)|

Σε όλα αυτά τα σενάρια, ο συγχρονισμός επιτυγχάνεται και βασίζεται πάνω σε γεγονότα. Στην συγκεκριμένη περίπτωση, το γεγονός βάση του οποίου έγινε ο συγχρονισμός ήταν η τοποθέτηση των γυαλιών από έναν ηθοποιό. Βέβαια, για να δημιουργηθεί η σύνδεση ήταν αναγκαίο να γνωρίζουμε εκ των προτέρων τόσο το περιεχόμενο της ταινίας όσο και της διαφήμισης.

Το σύστημα που θα υλοποιήσουμε θα μπορεί να εξυπηρετήσει και τέτοιες περιπτώσεις με πολυμέσα που το περιεχόμενό τους θα είναι γνωστό εκ των προτέρων. Έτσι, για παράδειγμα, θα δείξουμε πως σε έναν ποδοσφαιρικό αγώνα που μεταδίδεται ζωντανά, θα μπορούμε να παρουσιάσουμε και στοιχεία

που είναι γνωστά εκ των προτέρων όπως η βαθμολογία των ομάδων ή παλαιά στιγμιότυπα άλλων ποδοσφαιρικών αγώνων.

### Live generated content

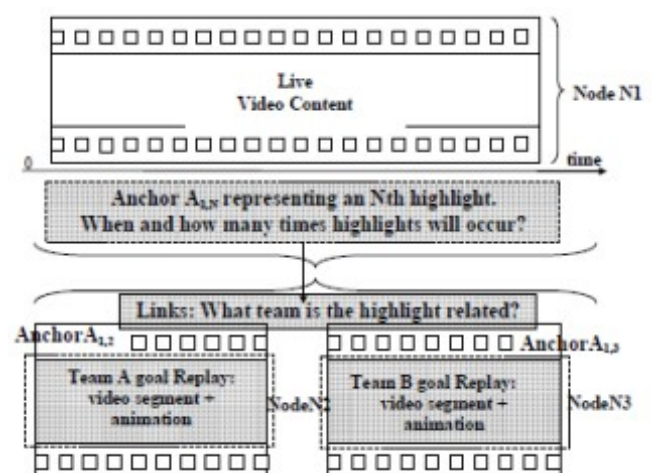
Αντίθετα με την προηγούμενη περίπτωση, στο σενάριο μιας ζωντανής πολυμεσικής παρουσίασης το περιεχόμενο δεν είναι γνωστό εκ των προτέρων. Το αποτέλεσμα είναι ότι δεν διαθέτουμε σίγουρα γεγονότα πάνω στα οποία ο συγχρονισμός μπορεί να οριστεί εύκολα με την δημιουργία των αντίστοιχων anchors.

Χαρακτηριστικά παραδείγματα αυτής της κατηγορίας αποτελούν ένας ποδοσφαιρικός αγώνας, η τελετή απονομής των Oscar και γενικά οποιαδήποτε πολυμεσική παρουσίαση που διεξάγεται ζωντανά. Το περιεχόμενο τέτοιων παρουσιάσεων αποτελείται από βίντεο, ήχο αλλά και έξτρα περιεχόμενο. Το έξτρα περιεχόμενο συνήθως αφορά πληροφορίες που είναι γνωστές πριν από την έναρξη της μετάδοσης, όπως για παράδειγμα οι συνθέσεις των ποδοσφαιρικών αγώνων ή οι υποψηφιότητες για κάποιο βραβείο Oscar.

Σε όλες αυτές τις περιπτώσεις, γνωρίζουμε ότι κάποια γεγονότα ενδεχομένως να συμβούν αλλά δεν γνωρίζουμε εκ των προτέρων πόσες φορές θα συμβούν αλλά και πότε θα συμβούν. Έτσι, σε έναν ποδοσφαιρικό αγώνα τέτοια γεγονότα μπορούμε να θεωρήσουμε την επίτευξη γκολ ή την κόκκινη κάρτα ή σε έναν αγώνα φόρμουλα 1 μπορούμε να ορίσουμε ως γεγονός την εκκίνηση, τον τερματισμό αλλά και τις συγκρούσεις μεταξύ αυτοκινήτων. Όλα τα παραπάνω γεγονότα είναι μεταβλητά και δεν μπορεί να προβλεφθεί η ακριβής χρονική στιγμή τέλεσής τους μέσα στην μετάδοση.

Για παράδειγμα, ας θεωρήσουμε τη ζωντανή μετάδοση ενός ποδοσφαιρικού αγώνα και ως γεγονός ας εξετάσουμε την επίτευξη ενός γκολ. Θέλουμε, μετά την επίτευξη του γκολ, να δημιουργήσουμε ένα βίντεο που θα δείχνει το γκολ (replay) όπως μεταδόθηκε ζωντανά καθώς και ένα βίντεο που θα δείχνει την μασκότ της ομάδας που πέτυχε το γκολ να πανηγυρίζει.

Όπως φαίνεται και στο σχετικό σχεδιάγραμμα, ο κόμβος N1 (Node N1) αντιπροσωπεύει την ζωντανή πολυμεσική παρουσίαση, το anchor  $A_{1,2}$  αντιπροσωπεύει τον κόμβο N2 (Node N2) που αναφέρεται σε ένα βίντεο με την μασκότ της 1<sup>ης</sup> ομάδας να πανηγυρίζει, ενώ αντίστοιχα το anchor  $A_{1,3}$  αντιπροσωπεύει τον κόμβο N3 (Node N3) που αναφέρεται σε ένα βίντεο με την μασκότ της 2<sup>ης</sup>



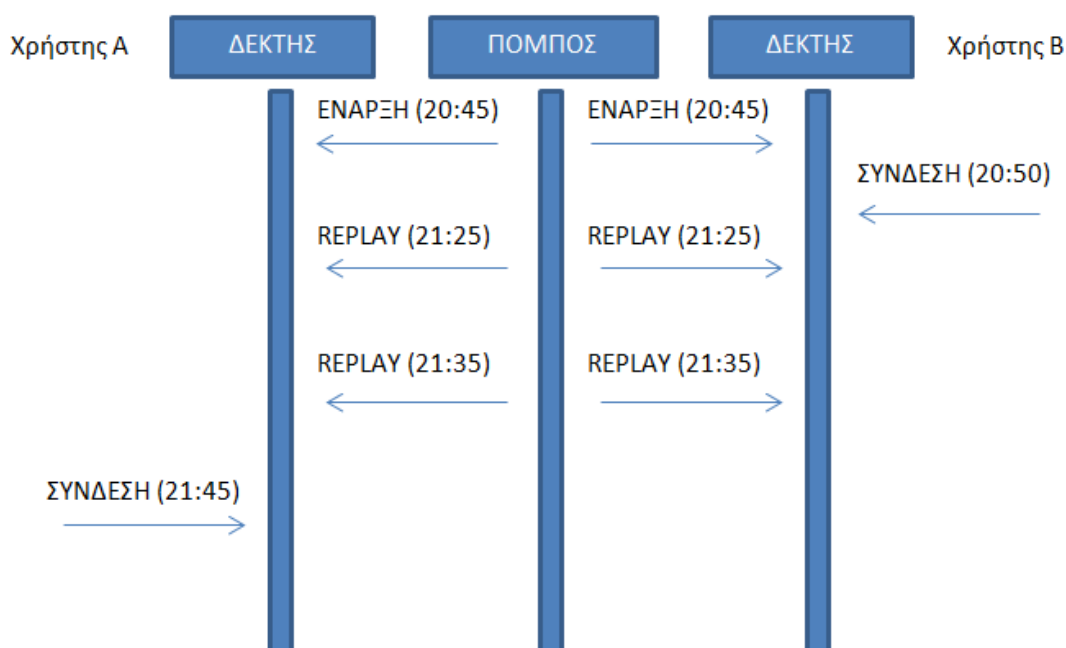
ομάδας να πανηγυρίζει. Και τα δύο βίντεο N2 και N3 αφορούν έξτρα περιεχόμενο που ήταν γνωστά εκ των προτέρων και έτσι μπορούσαν να οριστούν και τα αντίστοιχα anchors  $A_{1,2}$  και  $A_{1,3}$ .

Τα δυναμικά anchor  $A_{1,N}$  δημιουργούνται κάθε φορά που επιτυγχάνεται ένα γκολ. Ο συγχρονισμός επιτυγχάνεται μέσω των συνδέσεων link  $L(A_{1,N} - A_{1,2}$  ή  $A_{1,3})$ . Έτσι, ουσιαστικά δημιουργείται ένα replay που παρουσιάζει πρώτα το live generated content με τη βοήθεια του anchor  $A_{1,N}$  και μετά ένα εκ των δύο βίντεο N1 ή N2 με την βοήθεια των anchors  $A_{1,2}$  ή  $A_{1,3}$  αναλόγως την ομάδα που πέτυχε το γκολ.

Το σύστημα που θα υλοποιήσουμε θα ασχοληθεί με την διαχείριση τέτοιων καταστάσεων. Θα μελετήσουμε, δηλαδή, ζωντανές πολυμεσικές παρουσιάσεις, όπου βάση κάποιων γεγονότων που θα ορίζονται πρωτύτερα θα δημιουργούνται και τα αντίστοιχα replay. Σε αυτήν την περίπτωση των ζωντανών πολυμεσικών παρουσιάσεων παρουσιάζεται και το κύριο πρόβλημα που θα μας απασχολήσει στην παρούσα αναφορά, το πρόβλημα του Late tuning.

### Late tuning

Ένα ιδιαίτερο πρόβλημα που παρουσιάζεται στις παραπάνω περιπτώσεις είναι και το πρόβλημα που είναι γνωστό ως late tuning. Σε αυτό το σενάριο, ο θεατής δεν παρακολουθεί τη ζωντανή μετάδοση από την αρχή με αποτέλεσμα να χάσει κάποια σημαντικά γεγονότα, όπως για παράδειγμα την εκκίνηση σε ένα αγώνα φόρμουλα1 ή την επίτευξη ενός γκολ σε έναν ποδοσφαιρικό αγώνα.



Όπως φαίνεται και στο παραπάνω σχεδιάγραμμα, έχουμε δύο χρήστες (Α και Β) που ξεκινούν να παρακολουθούν μια ζωντανή μετάδοση από διαφορετική ώρα. Η μετάδοση ξεκινά στις 20:45, ο χρήστης Β συντονίζεται στις 20:50 ενώ ο χρήστης Α συντονίζεται με το ζωντανό πρόγραμμα στις 21:45.

Θεωρούμε ότι στην συγκεκριμένη μετάδοση συμβαίνουν δύο σημαντικά γεγονότα, το ένα στις 21:25 και το άλλο στις 21:35. Όπως είναι προφανές ο χρήστης Β βλέπει τα δύο αυτά γεγονότα ζωντανά ενώ ο χρήστης Α την στιγμή που συντονίζεται έχει χάσει και τα δύο αυτά γεγονότα. Αν δεν έχει προβλεφθεί η δημιουργία και μετάδοση replay τότε ο χρήστης Α δεν θα δει ποτέ αυτά τα γεγονότα.

Παρόμοιο πρόβλημα παρουσιάζεται και όταν πολλές ζωντανές μεταδόσεις λαμβάνουν χώρα την ίδια χρονική στιγμή. Χαρακτηριστικό παράδειγμα είναι η τελευταία αγωνιστική ενός ποδοσφαιρικού πρωταθλήματος όταν όλοι οι αγώνες διεξάγονται την ίδια ώρα για λόγους ισονομίας. Ο χρήστης που θα επιλέξει να παρακολουθήσει έναν αγώνα σε ζωντανή μετάδοση, θα πρέπει να έχει και τη δυνατότητα να παρακολουθήσει τα highlight από τους υπόλοιπους αγώνες.

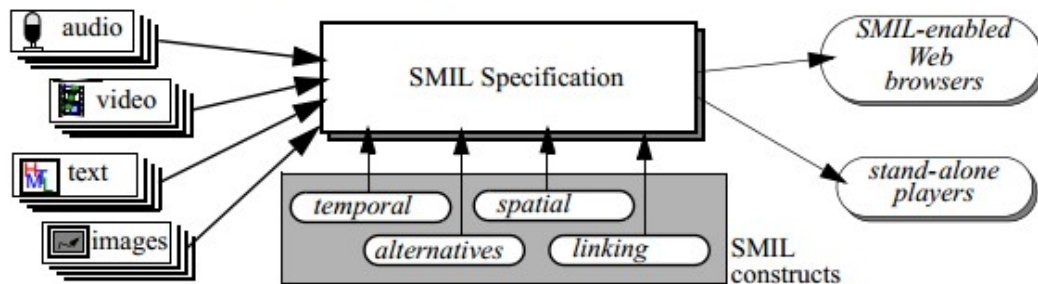
Το μοντέλο που θα υλοποιήσουμε θα πρέπει να δίνει στον θεατή τη δυνατότητα να βλέπει αυτά τα γεγονότα που έχασε μέσω ενός διαδραστικού λογισμικού. Ο πομπός θα πρέπει να δημιουργεί το replay του σημαντικού γεγονότος και να το μεταδίδει ώστε οι χρήστες να μπορούν να το δουν μετά από ορισμένη ώρα.

## 2.3 ΠΡΟΤΥΠΟ SMIL

Η γλώσσα SMIL αποτελεί πρότυπο για την δημιουργία πολυμεσικών παρουσιάσεων και δημιουργήθηκε από την W3C το 1998 [14]. Πρόκειται για μια δηλωτική γλώσσα που χρησιμοποιείται κυρίως σήμερα για να περιγράψει διαδικτυακά πολυμεσικά αντικείμενα που προβάλλονται είτε σε κάποιον browser είτε σε stand-alone εφαρμογές. Από την πρώτη της εμφάνιση μέχρι σήμερα, έχουν παρουσιαστεί αρκετές εφαρμογές διαχείρισης αρχείων SMIL που προσφέρουν στον χρήστη διάφορες δυνατότητες όπως συντάκτες αρχείων, εφαρμογές προεπισκόπησης αλλά και εφαρμογές αναπαραγωγής SMIL αρχείων. Έχει χρησιμοποιηθεί επιτυχώς σε μεγάλα συστήματα, όπως στην περιγραφή της MMS (Multimedia Message System) αλλά και στην βραζιλιάνικη ψηφιακή τηλεόραση για θέματα αλληλεπίδρασης και συγχρονισμού [2].

Εξαιρετικά ενδιαφέρονσα παρουσιάζεται η χρήση της αλλά και η μελλοντική της προοπτική στον κόσμο του διαδικτύου. Παρόλο που η SMIL δεν δημιουργήθηκε με σκοπό να αντικαταστήσει την εφαρμογή του Adobe Flash, παρουσιάζεται ως μια καλή εναλλακτική για τον συγχρονισμό πολυμεσικών αντικειμένων που περιέχονται μέσα σε διαδικτυακές σελίδες [2].

Η γλώσσα SMIL περιγράφει κυρίως τη χρονική συμπεριφορά όλων των αντικειμένων σε μια πολυμεσική παρουσίαση, όπως βίντεο, ήχος, εικόνες, κείμενο κτλ. Δεν ασχολείται καθόλου με τα περιεχόμενα της πολυμεσικής παρουσίασης και των αντικειμένων που την συγκροτούν, αλλά περιγράφει πως αυτά τα αντικείμενα συνδυάζονται [3]. Η αρχιτεκτονική της παρουσιάζεται στο παρακάτω σχήμα:



Όπως φαίνεται και παραπάνω, η δομή του προτύπου SMIL περιγράφει τέσσερα βασικά στοιχεία μιας πολυμεσικής παρουσίασης:

- 1) **Χρονικές προδιαγραφές:** Προσδιορίζει τη χρονική συμπεριφορά των στοιχείων της παρουσίασης, όπως την έναρξη, τη λήξη, την διάρκεια κτλ.
- 2) **Χωρικές προδιαγραφές απεικόνισης:** Δημιουργεί αρχέτυπα για την διάταξη των αντικείμενων μέσα σε μια παρουσίαση.
- 3) **Εναλλακτικές προδιαγραφές συμπεριφοράς:** Ορίζει παραμέτρους που βασίζονται σε διαφορετικές απαιτήσεις χρηστών ή συστημάτων.
- 4) **Υποστήριξη υπερμέσων:** Δημιουργεί μηχανισμούς για την σύνδεση διάφορων μερών των παρουσιάσεων (anchors και links).

Μια παρουσίαση μπορεί εύκολα να προσδιοριστεί χρονικά (timeline) [4] αλλά και χωρικά με την βοήθεια της SMIL. Είναι, δηλαδή, υπεύθυνη για τον χώρο και τον τρόπο προβολής αυτών των αντικειμένων μέσα στην παρουσίαση και σε καμία περίπτωση δεν αντικαθιστά τους άλλους τύπους αρχείων, όπως το MPEG για τις εφαρμογές βίντεο ή το HTML για το κείμενο.

Ο συγχρονισμός των αντικειμένων γίνεται είτε παράλληλα, είτε σε ακολουθία είτε με ορισμό επιλογών. Η σύνθεση γίνεται πάντοτε ιεραρχικά και το γεγονός αυτό προσφέρει στη γλώσσα SMIL τη δυνατότητα οποιαδήποτε παρουσίαση να προσαρμόζεται εύκολα και ικανοποιητικά σε διαφορετικές συνθήκες. Έτσι, για παράδειγμα, μια πολυμεσική παρουσίαση που δημιουργήθηκε για το διαδίκτυο

μπορεί να προσαρμοσθεί σε συστήματα όπως ένα κινητό [4], μεταβάλλοντας χαρακτηριστικά της όπως το bandwidth για να αποδώσει καλύτερα.

Υπάρχουν τρεις κύριοι μηχανισμοί μέσω των οποίων η SMIL δίνει την δυνατότητα διαχείρισης και αλληλεπίδρασης σε μια πολυμεσική παρουσίαση: μια γενική αρχιτεκτονική συνδεσμολογίας μέσω anchors που αναφέρονται σε συγκεκριμένα χρονικά σημεία της παρουσίασης και είδαμε στην προηγούμενη ενότητα, μέσω γεγονότων και τέλος μέσω ασύγχρονων προγραμματισμών [1].

Ο συγχρονισμός των αντικειμένων μέσω γεγονότων γίνεται με βάση δύο τύπους γεγονότων: τα εσωτερικά που είναι ήδη ορισμένα κατά την δημιουργία της παρουσίασης και τα εξωτερικά που προκαλούνται από χρήστες μέσω συγκεκριμένων διεπαφών ( User Interfaces). Στην πρώτη περίπτωση, ένα εσωτερικό γεγονός όπως το τέλος ενός βίντεο πυροδοτεί την έναρξη ενός άλλου βίντεο. Στην δεύτερη περίπτωση του εξωτερικού γεγονότος, ο χρήστης μπορεί να διαδραστικά να παρέμβει στην ροή μιας παρουσίασης, πατώντας ένα κουμπί ή ένα σύνδεσμο.

Παρότι, λοιπόν, υπάρχουν διάφοροι μηχανισμοί που επιτρέπουν την διαδραστικότητα σε μια παρουσίαση, δυστυχώς, η γλώσσα SMIL δεν προσφέρει από μόνη της στον χρήστη παραμέτρους για να αλλάξει δυναμικά μια πολυμεσική παρουσίαση και να επηρεάσει τη μελλοντική της συμπεριφορά. Επιπλέον, τα γεγονότα μέσα στην παρουσίαση είναι ο μόνος τρόπος επικοινωνίας με τον χρήστη και δεν μπορεί να μεταφέρει άλλου είδους δεδομένα στον εξωτερικό κόσμο. Τα δύο αυτά μειονεκτήματα καθιστούν την SMIL ανεπαρκή από μόνη της να συγχρονίσει επιτυχώς το περιεχόμενο μιας ζωντανής μετάδοσης και να δώσει λύση στα προβλήματα Late tuning.

### **3. ΣΧΕΤΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ**

#### **3.1 ΠΑΡΟΥΣΙΑΣΗ**

Το πρόβλημα της διαχείρισης και προβολής πολυμεσικών παρουσιάσεων στο διαδίκτυο παρουσιάστηκε από τα πρώτα χρόνια της εμφάνισης του διαδικτύου. Αρχικά, ακόμα και απλές στατικές εικόνες προβάλλονταν σε ξεχωριστά παράθυρα γιατί ο συντονισμός τους με απλό κείμενο ήταν μια δύσκολη υπόθεση [11]. Η εμφάνιση δηλωτικών προτύπων όπως η SMIL, η SVG (Scalable Vector Graphics) αλλά και η HTML5 αργότερα επέτρεψαν την ομαλή εισαγωγή πολυμεσικών αντικειμένων στο διαδίκτυο και όρισαν τον χρόνο ως τη βασική προτεραιότητα σε πολυμεσικές παρουσιάσεις. Παράλληλα, η εξάπλωση και εξέλιξη των γλωσσών προγραμματισμού που χρησιμοποιούνται στο διαδίκτυο επέτρεψαν την εισαγωγή της έννοιας της διαδραστικότητας σε αυτές τις παρουσιάσεις.

Παραδοσιακά, η προσέγγιση της διαχείρισης των πολυμεσικών παρουσιάσεων στη σχετική βιβλιογραφία γίνεται με τους εξής δύο τρόπους:

1. **Διαδικαστική ή προστακτική προσέγγιση** (procedural or imperative approach). Η προσέγγιση αυτή έχει ως βάση της μια γλώσσα προγραμματισμού, όπως JavaScript ή Action Script, και χαρακτηριστικό παράδειγμα είναι η εφαρμογή Macromedia Flash [4]. Ο χρήστης που έχει το απαραίτητο υπόβαθρο γνώσεων μπορεί να δημιουργήσει πολυμεσικές εφαρμογές που θα προσφέρουν το πλεονέκτημα της αλληλεπίδρασης με τον θεατή.
2. **Δομημένη ή δηλωτική προσέγγιση** (structured or declarative approach). Η προσέγγιση αυτή έχει ως βάση της μια δηλωτική markup γλώσσα όπως HTML, CSS ή το πρότυπο SMIL. Οι πολυμεσικές παρουσιάσεις που δημιουργούνται με αυτόν τον τρόπο έχουν ως βάση τους τον χρόνο και επιτυγχάνεται συγχρονισμός μεταξύ των αντικειμένων που απαρτίζουν τη πολυμεσική παρουσίαση.

Μελετώντας κανείς την βιβλιογραφία φαίνεται ότι αρχικά είχε επικρατήσει η διαδικαστική προσέγγιση όπου ο δημιουργός της παρουσίασης έπρεπε να φροντίσει για την παρουσίαση και την προσαρμογή της πολυμεσικής εφαρμογής μέσω scripting. Αντίθετα, πρότυπα όπως η SMIL, αναφέρονταν μόνο θεωρητικά και δεν είχαν ενεργή συμμετοχή στην υλοποίηση τέτοιων εφαρμογών. Δεν θα αναφερθούμε λεπτομερώς σε αυτά τα μοντέλα γιατί θεωρούνται πλέον ξεπερασμένα και δεν μπορούν να αντιμετωπίσουν τα προβλήματα συγχρονισμού σε ζωντανές πολυμεσικές μεταδόσεις.

Με το πέρασμα του χρόνου και την εισαγωγή και άλλων δομημένων προτύπων η σχετική βιβλιογραφία δείχνει να στρέφεται στην δηλωτική προσέγγιση. Ένα τέτοιο νέο πρότυπο, βασισμένο στην τεχνολογία της XForms που αναπτύξαμε σε προηγούμενο κεφάλαιο, είναι και η γλώσσα XBL (Xml binding language). Η γλώσσα XBL βασίζεται στην γλώσσα XML και μπορεί να εμπλουτίσει τη συμπεριφορά των απλών HTML κειμένων στις διαδικτυακές παρουσιάσεις [7]. Οι εμπνευστές της XBL προσπάθησαν να χρησιμοποιήσουν μια πιο δομημένη λογική που θα αντικαθιστούσε τη λογική και την χρήση των γλωσσών προγραμματισμού. Ο δημιουργός μιας πολυμεσικής παρουσίασης μπορεί μέσω της XBL να ορίσει ορισμένα νέα χαρακτηριστικά βάση κάποιων παραμέτρων και να κάνει μια παρουσίαση εν μέρει δυναμική. Ακολούθως, ήρθε να προστεθεί η τεχνολογία XConnector που επέκτεινε την XBL, προσθέτοντας την βασική παράμετρο του χρόνου. Για παράδειγμα, ο δημιουργός της παρουσίασης μπορούσε να ορίσει ότι ένα βίντεο μπορεί να ξεκινήσει όταν ένα άλλο βίντεο τελειώσει μέσα σε μια παρουσίαση.

Τελευταία, μερικά μοντέλα προσπάθησαν να θεωρήσουν τον χρόνο ως το κύριο συστατικό μιας πολυμεσικής εφαρμογής, χρησιμοποιώντας markup γλώσσες όπως η SMIL (Synchronized Multimedia Integration Language) και η NCL (Nested Context Language). Η γλώσσα SMIL 2.0 που θα μας απασχολήσει περισσότερο στη συνέχεια, προσφέρει πλήρη υποστήριξη όσον αφορά την χρονική



διαχείριση και τον έλεγχο αλληλεπίδρασης σε μια κατανεμημένη πολυμεσική παρουσίαση. Η υποστήριξη αυτή αναφέρεται κυρίως στις δυνατότητες που δίνονται στον δημιουργό της παρουσίασης να δημιουργήσει εύκολα συγχρονισμένες, πολύπλοκες πολυμεσικές παρουσιάσεις χωρίς μάλιστα να χρειάζεται να έχει ιδιαίτερες γνώσεις προγραμματισμού.

Δυστυχώς, όμως, το πρότυπο SMIL από μόνο του δεν μπορεί να υποστηρίξει την πλήρη υλοποίηση και παρουσίαση μιας πολυμεσικής παρουσίασης στο διαδίκτυο. Η συνύπαρξη μιας διαδικτυακής γλώσσας προγραμματισμού, όπως η JavaScript, μπορεί να βοηθήσει να ξεπεραστούν θέματα όπως η απόδοση της παρουσίασης αλλά και η δυνατότητα αλληλεπίδρασης με τον απλό χρήστη. Ειδικά, η μεγάλη δυνατότητα της JavaScript είναι, ότι σε αντίθεση με τον Adobe Flash που χρησιμοποιείται ευρέως, μπορεί να βελτιώσει την προσιτότητα του διαδικτύου στον απλό χρήστη [2]. Ο τελευταίος μπορεί να επιλέξει εναλλακτικούς τρόπους παρουσίασης ανάλογα με τα ενδιαφέροντά του, όπως για παράδειγμα την επιλογή κατάλληλων υποτίτλων για μια παρουσίαση.

Στα πλαίσια αυτής της λογικής, οι Scherp και Boll παρουσίασαν για πρώτη φορά μια πολυμεσική εφαρμογή που ήταν προσωποποιημένη και παραμετροποιημένη στο προφίλ του χρήστη [5]. Συγκεκριμένα, παρουσίασαν έναν πολυμεσικό τουριστικό οδηγό για κινητά με το οποίο ο χρήστης μπορούσε να αλληλεπιδράσει και να δημιουργήσει τις προσωπικές του εφαρμογές. Δημιούργησαν ένα ανεξάρτητο λογισμικό που προσαρμοζε την εφαρμογή όχι μόνο στις προτιμήσεις του χρήστη αλλά και στις δυνατότητες της συσκευής που χρησιμοποιούσε ο χρήστης.

Σε συνέχεια της προηγούμενης λογικής, οι Bulterman και Jansen πρότειναν μια νέα τεχνολογία που ονόμασαν SMIL State και συνδυάζει πρότυπα γλώσσες που βασίζονται στον χρόνο, όπως η SMIL με εξωτερικά μοντέλα δεδομένων [4]. Το σύστημα που πρότειναν επιτρέπει την δήλωση και χρήση παραμέτρων σε πολυμεσικές παρουσιάσεις. Το μοντέλο δεδομένων είναι ανεξάρτητο από το υπόλοιπο σύστημα, γεγονός που το κάνει εύκολα προσβάσιμο από άλλες διαδικτυακές εφαρμογές. Μάλιστα, το μοντέλο του SMIL State προτάθηκε στο SYMM Working Group και έγινε δεκτό να συμπεριληφθεί στο πρότυπο SMIL 3.0.

Τέλος, εμφανίστηκαν και μερικά μοντέλα που προσπάθησαν να συνδυάσουν πολλά διαφορετικά πρότυπα σε μια πλήρως δηλωτική προσέγγιση. Η πιο επιτυχημένη από αυτές τις προσπάθειες, με το παραπλανητικό όνομα Timesheets.js, συνένωσε τις δυνάμεις των προτύπων HTML5, CCS3 και SMIL [11] με σκοπό οι διαδικτυακές πολυμεσικές παρουσιάσεις να έχουν ως κύριο άξονά τους τον χρόνο. Μάλιστα, στο συγκεκριμένο μοντέλο χρησιμοποιήθηκε η τελευταία έκδοση της SMIL 3.0 που πρόσφερε νέες δυνατότητες. Οι νέες αυτές δυνατότητες που σκόπευαν στον καλύτερο συγχρονισμό περιελάμβαναν μεταξύ άλλων την SMIL State, που αναφέραμε προηγουμένως, αλλά και την τεχνολογία SMIL Transitions. Βέβαια, και στην υλοποίηση του συγκεκριμένου μοντέλου, έπαιξε ρόλο

η γλώσσα προγραμματισμού JavaScript. Το τελικό αποτέλεσμα ήταν μια βιβλιοθήκη JavaScript που είναι ανοιχτού κώδικα και μπορεί να λειτουργήσει σε οποιονδήποτε browser.

Σε αντίθεση με τις προηγούμενες προσεγγίσεις που αναφέρονταν κυρίως σε περιπτώσεις παρουσιάσεων που το περιεχόμενο ήταν γνωστό εκ των προτέρων, εμφανίστηκαν και δηλωτικές προσεγγίσεις για το συγχρονισμό σε ζωντανές πολυμεσικές παρουσιάσεις. Έτσι, αρχικά, παρουσιάστηκε ένα γενικό θεωρητικό μοντέλο που ενσωμάτωνε οντότητες πολυμέσων μέσα σε διαδικτυακές σελίδες που επέτρεπαν την ανταλλαγή δεδομένων μεταξύ ενός πομπού και των δεκτών και ενός θεωρητικού μοντέλου pull or push στο διαδίκτυο [9]. Η βασική ιδέα ήταν ότι οι ρυθμίσεις γίνονταν αποκλειστικά από τον πομπό που μπορούσε να παραμετροποιήσει και να αλληλεπιδράσει με την πολυμεσική παρουσίαση.

Σε μια τέτοια προσπάθεια δημιουργίας ενός ολοκληρωμένου, και όχι μόνο θεωρητικού, μοντέλου που θα δίνει τη λύση στο πρόβλημα του late tuning, χρησιμοποιήθηκε η αντικειμενοστραφής περιγραφή για τα σημεία μιας ζωντανής παρουσίας [8]. Οι Neto και Santos παρουσίασαν μια διαδικαστική προσέγγιση βασισμένη στα γεγονότα που συμβαίνουν σε ένα αλληλεπιδραστικό σενάριο και επικεντρώθηκαν στις περιπτώσεις των ζωντανών show στην ψηφιακή τηλεόραση IDTV (Interactive Live TV Shows). Σε αυτές τις παρουσιάσεις, ο συγχρονισμός και η διαχείριση του χρόνου αποτελούσαν κλειδί για μια σωστή αλληλεπιδραστική παρουσίαση που περιέχει σίγουρα βίντεο και ήχο.

Οι Neto και Santos προσπάθησαν να ορίσουν ποια γεγονότα μπορεί να συμβούν κατά την διάρκεια της ζωντανής παρουσίας χωρίς να γνωρίζουν πότε θα συμβούν αυτά. Ο συγχρονισμός επιτεύχθηκε με τη βοήθεια δημιουργίας χρονικών συνδέσμων (anchors) που φυσικά δεν ήταν γνωστά πριν από την έναρξη της ζωντανής μετάδοσης. Με βάση αυτήν τη λογική, τα διάφορα πολυμέσα παρουσιάστηκαν ως αντικείμενα ενώ οι σχέσεις συγχρονισμού μεταξύ τους μοντελοποιήθηκαν ως μηνύματα που ανταλλάσσουν τα αντικείμενα αυτά. Τα μηνύματα από τον broadcaster στον viewer γίνονταν είτε μέσω της τεχνολογίας carousel DSM-CC που ανανεωνόταν με την προσθήκη ενός νέου γεγονότος (highlight) ή μέσω ενός μοντέλου push or push/pull όπου τα νέα γεγονότα μέσω ενός content provider ήταν προσβάσιμα στον viewer.

Για τις ανάγκες της πειραματικής αξιολόγησης, υλοποίησαν έναν εξομοιωτή για IDTV που το ονόμασαν LiveTV Simulator και εξομοίωνε ένα ψηφιακό τηλεοπτικό σταθμό σε ένα δίκτυο υπολογιστών. Έτσι, η κύρια μετάδοση γινόταν μέσω του πρωτοκόλλου RTP (Real-time Transport Protocol) και η μετάδοση της έξτρα πληροφορίας μέσω συνδέσεων TCP. Η πλατφόρμα LiveTV Simulator επέτρεπε τον διαχειριστή της να επεξεργαστεί τη ζωντανή μετάδοση για να δημιουργήσει νέα πολυμεσικά αρχεία. Από την άλλη, δημιουργήθηκε μια εφαρμογή σε γλώσσα

προγραμματισμού Java που επέτρεπε στον χρήστη μέσα από ένα διαδραστικό μενού να προσπελάσει και να προβάλλει τα highlights που δημιουργήθηκαν.

### 3.2 ΚΡΙΤΙΚΗ

Όπως αναφέραμε και στην προηγούμενη ενότητα, η προσέγγιση στο πρόβλημα της διαχείρισης των πολυμεσικών παρουσιάσεων ποικίλλει στην βιβλιογραφία με αποτέλεσμα να ποικίλουν και οι αντίστοιχες τεχνολογίες που χρησιμοποιούνται. Σκοπός της παρούσας ενότητας είναι να μελετήσουμε τα μειονεκτήματα και τα πλεονεκτήματα της κάθε τεχνολογίας όπως αυτά προκύπτουν από τα μοντέλα που παρουσιάσαμε για να δούμε ποια τεχνολογία μπορεί να χρησιμοποιηθεί για το σύστημα που θέλουμε να υλοποιήσουμε.

Γενικά, θα συγκρίνουμε γλώσσες προγραμματισμού (JavaScript, Action Script) με τα πρότυπα SMIL αλλά και τα πρότυπα που κυριαρχούν σήμερα στο διαδίκτυο (XHTML, HTML5, CSS3). Πριν αναφερθούμε στην κριτική της κάθε περίπτωσης, θα ορίσουμε τα χαρακτηριστικά που μας ενδιαφέρουν να ικανοποιεί το σύστημά μας και είναι τα ακόλουθα:

- **Structured/Declarative.** Πρέπει το σύστημα να είναι δομημένο έτσι ώστε να μπορούν να προβλεφθούν και να αντιμετωπιστούν εύκολα μελλοντικές συμπεριφορές.
- **Time-based.** Η υλοποίηση του συστήματος πρέπει να γίνει με βάση τεχνολογίες που έχουν ως βάση τον χρόνο και ως πρώτο μέλημα το συγχρονισμό.
- **Interactivity.** Το μοντέλο πρέπει να παρέχει παραμέτρους έτσι ώστε ο χρήστης να μπορεί να αλληλεπιδράσει με το σύστημα και να επηρεάσει τη ροή μιας παρουσίασης.
- **Adaptability.** Το σύστημα πρέπει να μπορεί να προσαρμόζεται ώστε το περιεχόμενο να είναι αποδοτικό σε διαφορετικές συσκευές και κάτω από διαφορετικές συνθήκες.
- **Accessibility.** Το περιεχόμενο της παρουσίασης πρέπει να μπορεί να γίνει αντικείμενο αναζήτησης μέσα στο διαδίκτυο.
- **Reusability.** Το μοντέλο πρέπει να είναι παραμετροποιημένο ώστε να μπορεί να χρησιμοποιηθεί σε πολλές περιπτώσεις διαφορετικών πολυμεσικών παρουσιάσεων.
- **Temporal links.** Το χαρακτηριστικό αυτό αναφέρεται στην δυνατότητα να μπορεί το σύστημα να αναφέρεται σε ένα συγκεκριμένο χρονικό σημείο μιας παρουσίασης ή ενός συγκεκριμένου πολυμέσου.
- **Scriptability.** Παρότι το σύστημα θα είναι δηλωτικού χαρακτήρα και θα βασίζεται σε πρότυπα, θα πρέπει να παρέχεται και η δυνατότητα να ξεπερνιούνται οι περιορισμοί αυτών των προτύπων. Αυτή η δυνατότητα υλοποιείται μέσω scripts, όταν εμφανίζονται ειδικές περιπτώσεις που δεν μπορεί να τις διαχειριστεί ένα πρότυπο.

- **Web rendering.** Εφόσον το διαδίκτυο θα είναι το τελικό εργαλείο που θα προβάλλει την πολυμεσική παρουσίαση, πρέπει το σύστημά μας να είναι συμβατό και αποδοτικό στους browsers του διαδικτύου.

Οι τεχνολογίες που επικράτησαν αρχικά όσον αφορά τις πολυμεσικές εφαρμογές στο διαδίκτυο ήταν οι γλώσσες προγραμματισμού. Συγκεκριμένα, χρησιμοποιήθηκε ευρέως η πλατφόρμα Flash, που είναι υλοποιημένη σε Action Script, αλλά και η γλώσσα JavaScript σε συνδυασμό με άλλες εφαρμογές αναπαραγωγής πολυμεσικών αρχείων [4]. Η χρησιμοποίηση του Flash, με πλέον χαρακτηριστικό παράδειγμα το YouTube, παραμένει ακόμα η πλέον διαδεδομένη. Έχει το πλεονέκτημα ότι είναι χρονικά προσανατολισμένη και παρέχει την δυνατότητα αλληλεπίδρασης με τον χρήστη. Ταυτόχρονα, όμως, η δυαδική της φύση κάνει το περιεχόμενο μη ανακτήσιμο και προσβάσιμο (accessibility) στον απλό χρήστη του διαδικτύου.

Η εναλλακτική της γλώσσας προγραμματισμού JavaScript είναι πιο εύκολα υλοποιήσιμη αλλά γενικά παρουσιάζεται λιγότερη αποτελεσματική από την τεχνολογία Flash. Παρέχει εξαιρετικές δυνατότητες παραμετροποίησης ώστε να δημιουργηθεί μια διαδραστική πολυμεσική παρουσίαση (reusability) και είναι αποδοτική στο διαδίκτυο (web rendering) σε συνδυασμό με ένα μέσο αναπαραγωγής πολυμεσικών αρχείων και με τη βοήθεια της HTML. Το κύριο πρόβλημά της είναι ότι δεν λαμβάνει υπόψη της τον χρόνο ως παράγοντα με αποτέλεσμα να έχει σοβαρά προβλήματα συγχρονισμού (time-based). Σήμερα, η λύση της αποκλειστικής χρήσης JavaScript προτείνεται κυρίως για μη δυναμικές, στατικές παρουσιάσεις που δεν έχουν ως κύρια βάση τους τον χρόνο. Συνοψίζοντας, οι τεχνολογίες Flash και JavaScript από μόνες τους αν και πρόσφεραν την αναγκαία παραμετροποίηση για να γίνει η παρουσίαση διαδραστική, δεν ήταν καθόλου δομημένες και υστερούσαν σε θέματα συγχρονισμού, ευκολίας στον χρήστη, προσαρμοστικότητας αλλά και πρόσβασης.

Για να διορθωθούν αυτά τα προβλήματα που παρουσιάζουν οι γλώσσες προγραμματισμού στις πολυμεσικές εφαρμογές, οι ερευνητές στράφηκαν στις δομημένες γλώσσες όπως το πρότυπο SMIL. Το κύριο πλεονέκτημα είναι ότι παρέχει ένα δομημένο περιβάλλον με κύριο σκοπό την διαχείριση του χρόνου. Το γεγονός αυτό συνεπάγεται ότι προσφέρει συγχρονισμό με τις δυνατότητες για προσαρμοστικότητα (adaptability) και προσβασιμότητα (accessibility). Επίσης, με την βοήθεια της SMIL υπάρχει η δυνατότητα για δημιουργία δυναμικών συνδέσμων σε συγκεκριμένα χρονικά σημεία μιας παρουσίασης (temporal links), γεγονός που θα μας βοηθήσει και στο πρόβλημα του late tuning.

Δυστυχώς όμως, παρά τις πολλές δυνατότητες που προσφέρει η γλώσσα SMIL για την δημιουργία μιας δυναμικής πολυμεσικής παρουσίασης, οι δημιουργοί τέτοιων παρουσιάσεων δείχνουν είτε να την αγνοούν είτε να μην είναι σε θέση να την εκμεταλλευτούν καταλλήλως. Αυτό οφείλεται, συνήθως, στο ότι η υποστήριξη από τις διαθέσιμες εμπορικές εφαρμογές είναι περιορισμένη (web rendering) με

αποτέλεσμα οι δημιουργοί τέτοιων παρουσιάσεων να παρουσιάζονται το λιγότερο σκεπτικοί [1,2]. Οι περισσότεροι Internet browsers δεν την υποστηρίζουν ενώ οι λίγες αυτόνομες εφαρμογές, όπως το Ambulant Player που ενσωματώνει και την τελευταία έκδοση 3.0 της SMIL είναι ανεπαρκής για να βοηθήσουν στην ενσωμάτωσή τους σε μια διαδικτυακή σελίδα.

Παράλληλα, η γλώσσα SMIL 2.0 παρουσίαζε μια καλύτερα δομημένη λύση αλλά υστερούσε στην παροχή επιλογών αλληλεπίδρασης στον χρήστη. Αυτό οφείλεται στο ότι η W3C που ανέπτυξε την SMIL, εστίασε περισσότερο στην ανάπτυξη της γλώσσας και δεν ασχολήθηκε ιδιαίτερα με την ανάπτυξη των αντίστοιχων εφαρμογών για την διαχείριση της. Έτσι, λοιπόν, διαπιστώθηκε σε συγκρίσεις που έγιναν ότι η παρουσία της SMIL είναι αναγκαία αλλά όχι και ικανή από μόνη της συνθήκη για να επιτευχθεί μια συγχρονισμένη πολυμεσική παρουσίαση [4].

Η εισαγωγή της δομημένης τεχνολογίας XForms προσπάθησε να αυξήσει τον βαθμό διαδραστικότητας στα δομημένα πρότυπα και να ελαχιστοποιήσει την χρήση των γλωσσών προγραμματισμού. Η XForms χρησιμοποίησε ένα μοντέλο δεδομένων που έκανε όλο το σύστημα εύκολα προσβάσιμο (accessibility), επαναχρησιμοποιήσιμο (reusability) και δυναμικό (interactivity). Παρότι η τεχνολογία αυτή δεν είναι βασισμένη στο χρόνο και δεν μπορεί να εφαρμοστεί σε διαδραστικές πολυμεσικές παρουσιάσεις, εντούτοις συνείσφερε στο γεγονός ότι χρησιμοποίησε μια δομημένη λογική. Έτσι, για παράδειγμα, το σύστημα έλεγχε την εγκυρότητα της τιμής εισόδου για τη μέρα της εβδομάδας χωρίς να χρησιμοποιεί JavaScript [6].

Μελλοντικές τεχνολογίες που αναπτύχθηκαν όπως η γλώσσα XBL παρότι έκαναν χρήση κάποιων παραμέτρων για να υλοποιήσουν μια παρουσίαση δυναμική δεν είχαν ως βάση τους το χρόνο και το συγχρονισμό αντικειμένων. Άλλες τεχνολογίες που παρουσιάστηκαν αργότερα, όπως η XConnector και η XHTML έβαλαν στο παιχνίδι και τον χρόνο με σκοπό να δημιουργήσουν διαδραστικές πολυμεσικές εφαρμογές αλλά και πάλι δεν μπόρεσαν να ενσωματώσουν το πρότυπο SMIL.

Συνοπτικά, παρουσιάζουμε τη σύγκριση των τεχνολογιών όσον αφορά τα χαρακτηριστικά που θέλουμε να έχει το σύστημά μας στον παρακάτω πίνακα:

	Flash	JavaScript	SMIL 2.0	XHTML+SMIL	SMIL 3.0	HTML5 + CSS3
<b>Structured</b>	-		+	+/-	+	+
<b>Time-based</b>	+		+	+	+	-
<b>Interactivity</b>	+	+	-	-	-	(limited)
<b>Adaptability</b>	+/-		+	+	+	+
<b>Accessibility</b>	-	+/-	+	+	+	+
<b>Reusability</b>	-	-	+/-	-	+	-
<b>Temporal links</b>	-	-	+	-	+	-
<b>Scriptability</b>	+	+	-	-	-	+
<b>Web rendering</b>	-	+	-	+	-	+

Αφού μελετήσαμε τα θετικά και τα αρνητικά των τεχνολογιών που αφορούν το θέμα των πολυμεσικών παρουσιάσεων στο διαδίκτυο, στη συνέχεια θα σχολιάσουμε τα κυριότερα μοντέλα που έχουν παρουσιαστεί στην βιβλιογραφία. Για κάθε μοντέλο θα αναφέρουμε τα θετικά τους στοιχεία αλλά και ποια προβλήματα παρουσιάζουν.

Αρχικά, η πολυμεσική εφαρμογή για κινητά που παρουσίασαν οι Scherp και Boll ήταν ένα πρώτο βήμα για την δημιουργία διαδραστικών συστημάτων με τα οποία ο χρήστης θα μπορεί να αλληλεπιδρά. Επίσης, το σύστημά τους ήταν από τα πρώτα που πρόσφεραν προσαρμοστικότητα (adaptability) στη συσκευή του χρήστη. Δυστυχώς, όλες οι έξτρα πολυμεσικές παρουσιάσεις που δημιουργούνταν ύστερα από πρωτοβουλία του χρήστη, δημιουργούνταν στον server και τον επιβάρυναν σημαντικά. Το κύριο μειονέκτημα του μοντέλου ήταν ότι παρότι στηριζόταν στη γλώσσα SMIL, δεν εκμεταλλευόταν πλήρως τις δυνατότητες της, με αποτέλεσμα να μην επιτυγχάνεται συγχρονισμός κάτω από πολύπλοκες συνθήκες.

Η προσπάθεια των Bulterman και Jansen με την υλοποίηση της πλατφόρμας SMIL State ήταν πιο ουσιαστική και έλυνε πολλά από τα προβλήματα συγχρονισμού και αλληλεπίδρασης στις πολυμεσικές εφαρμογές ακόμα και κάτω από πιο δύσκολες καταστάσεις. Το σύστημά τους αφορούσε ένα διαδικτυακό, πολυμεσικό τουριστικό οδηγό για μια συγκεκριμένη πόλη, ο οποίος προσαρμοζόταν τόσο στις επιθυμίες του χρήστη όσο και στις δυνατότητες που είχε το μηχάνημα που χρησιμοποιούσε. Οι νέες υπηρεσίες που προσφέρονταν στον χρήστη, σε αντίθεση με την πλατφόρμα των Scherp και Boll δεν επιβάρυναν τον server αλλά εκτελούνταν στο client μηχάνημα. Τα μεγαλύτερα πλεονεκτήματα που παρουσιάζει η πλατφόρμα SMIL State είναι τα ακόλουθα:

- **Device Independence:** Η πλατφόρμα έχει την δυνατότητα να προσαρμόσει το περιεχόμενό της αναλόγως των χαρακτηριστικών του συστήματος του χρήστη, όπως το bandwidth και το μέγεθος της οθόνης.

- **Reusability:** Επειδή το σύστημα βασίζεται σε ένα πλήρως παραμετροποιημένο μοντέλο δεδομένων, μπορεί να διαχειριστεί επαρκώς παρόμοιες πολυμεσικές παρουσιάσεις, χωρίς να παρέμβει σημαντικά ο δημιουργός τους.
- **Accessibility:** Το περιεχόμενο των παρουσιάσεων μπορεί να αναζητηθεί στο διαδίκτυο μέσω των μηχανών αναζήτησης (search engines).

Όμως, το μοντέλο δεδομένων που χρησιμοποιεί η πλατφόρμα SMIL State δεν είναι πλήρως παραμετροποιημένο. Οι συγγραφείς υποστηρίζουν ότι η πλατφόρμα τους μπορεί να διαχειριστεί και άλλες παρόμοιες παρουσιάσεις, όπως ένα τουριστικό οδηγό για άλλη πόλη, αλλά δεν είναι τόσο παραμετροποιημένο ώστε να μπορεί να διαχειριστεί άλλες διαφορετικού τύπου παρουσιάσεις. Μάλιστα, οι ίδιοι υποστηρίζουν ότι η ιδέα να προσθέσουν το πρότυπο SMIL στις πολυμεσικές παρουσιάσεις βασιζόταν σε παρουσιάσεις όπως τα ηλεκτρονικά παιχνίδια, τα quiz στο διαδίκτυο αλλά και η ηλεκτρονική εκπαίδευση. Δυστυχώς, η πλατφόρμα SMIL State δεν μπορεί να εφαρμοστεί σε ζωντανές παρουσιάσεις και ως εκ τούτου δεν συνεισφέρει στο πρόβλημα του Late Tuning.

Παράλληλα, η δημιουργία μιας διαδραστικής πολυμεσικής παρουσίασης με την SMIL αποδεικνύεται για τους περισσότερους μια δύσκολη αποστολή. Οι δημιουργοί τέτοιων παρουσιάσεων δείχνουν να μην εμπιστεύονται τα υπάρχοντα εργαλεία στην αγορά και να διστάζουν καν να ανοίξουν το μεγάλο κεφάλαιο του συγχρονισμού. Επιπλέον, το μεγάλο μειονέκτημα της SMIL αποτελεί ότι από μόνη της δεν προσφέρει παραμετροποίηση και άρα δεν μπορεί να συνεισφέρει στην δημιουργία μιας διαδραστικής πολυμεσικής παρουσίασης. Ακόμα και με τον συνδυασμό διαφορετικών προτύπων, όπως έγινε στην περίπτωση του μοντέλου Timesheets.js που συνδυάστηκαν SMIL, HTML5 και CSS3, το τελικό μοντέλο που υλοποιήθηκε ήταν μια βιβλιοθήκη γραμμένη σε JavaScript. Ο λόγος ήταν ότι οι browsers δεν μπορούσαν να υποστηρίξουν τις δυνατότητες της τελευταίας έκδοσης SMIL 3.0 και χρειαζόταν μια ενδιάμεση πλατφόρμα γραμμένη με scripting.

Επειδή, όμως, οι σύγχρονοι περιορισμοί οφείλονται περισσότερο στην ετερογένεια των προφίλ των διαδικτυακών χρηστών και στους πολλούς browser που υπάρχουν στην αγορά και όχι σε περιορισμούς στην γλώσσα SMIL, πιστεύουμε ότι στο μέλλον η διείσδυση της SMIL στην διαχείριση των πολυμέσων στο διαδίκτυο θα αυξηθεί σημαντικά. Παράλληλα, η επικράτηση διαδικτυακών προτύπων παρόμοιας λογικής και δομής όπως η XHTML και η HTML5 θα οδηγήσει τελικά σε ένα νέο SMIL πρότυπο που θα μπορεί εύκολα να ενσωματωθεί στις τρέχουσες εμπορικές εφαρμογές. Πιστεύουμε ότι ένα τέτοιο πρότυπο θα παίξει ένα σημαντικό ρόλο στο να αποτελέσει την κοινή βάση για διαδικτυακές παρουσιάσεις που θα εκτελούνται από τους περισσότερους browser. Κατόπιν, θα είναι ευθύνη της

κάθε εφαρμογής να εκμεταλλευτεί την SMIL αλλά και να τη βελτιώσει σε ποιότητα και αποδοτικότητα.

Ο συνδυασμός της SMIL με άλλες τεχνολογίες όπως η XForms, η XBL αλλά και η XHTML προσφέρει κάποιες νέες δυνατότητες. Η αλληλεπίδραση με τον χρήστη αυξάνεται σημαντικά αλλά παράλληλα αυξάνεται και ο κίνδυνος να χαθεί η χρονική δόμηση και ο συγχρονισμός που προσφέρει το πρότυπο SMIL.

Όλα τα συστήματα που παρουσιάσαμε μέχρι τώρα αναφέρονται κυρίως σε πολυμεσικές παρουσιάσεις που το περιεχόμενο είναι γνωστό εκ των προτέρων. Με άλλα λόγια, τα μοντέλα αυτά δεν μπορούν να εφαρμοστούν σε παρουσιάσεις με live generated content και άρα δεν αφορούν το πρόβλημα του Late tuning. Θυμίζουμε ότι το σενάριο του Late tuning παρουσιάζεται σε ζωντανές, πολυμεσικές παρουσιάσεις όπου τόσο ο συγχρονισμός αλλά και η αλληλεπίδραση είναι τα βασικά συστατικά. Αυτό το γεγονός καθιστά την δηλωτική προσέγγιση ως την καταλληλότερη μέθοδο μιας και αλλιώς ο συγχρονισμός και η αλληλεπίδραση δεν γίνεται να επιτευχθούν. Κυρίαρχο ρόλο σε αυτήν την προσέγγιση παίζει η δηλωτική γλώσσα SMIL.

Οι Neto και Santos παρουσίασαν μια διαδικαστική προσέγγιση βασισμένη στα γεγονότα που συμβαίνουν σε ένα αλληλεπιδραστικό σενάριο και επικεντρώθηκαν στις περιπτώσεις των ζωντανών show στην ψηφιακή τηλεόραση IDTV (Interactive Live TV Shows). Δημιούργησαν έναν δικό τους εξομοιωτή που ονόμασαν LiveTVSimulator για να εξομοιώσουν τη μετάδοση της ψηφιακής τηλεόρασης μέσα από ένα δίκτυο υπολογιστών [8]. Επίσης, με τη βοήθεια της Java δημιούργησαν μια εφαρμογή μέσω της οποίας ο broadcaster μπορούσε να επιλέγει τα γεγονότα που θεωρούσε σημαντικά. Αντίστοιχα, δημιουργήθηκε μια εφαρμογή μέσω της οποίας ο χρήστης μπορούσε να επιλέξει και να προβάλει τα γεγονότα αυτά αργότερα ώστε να δοθεί λύση στο πρόβλημα του Late tuning.

Παρά την πειραματική επιτυχία του εγχειρήματος και τη μετάδοση των highlights χωρίς καθυστέρηση στον χρήστη, ο ρόλος της SMIL στο όλο μοντέλο ήταν περιορισμένος. Αντίθετα, το σύστημα δημιουργούσε πολυμεσικά αντικείμενα για κάθε highlight αν και δεν αναφερόταν πως γινόταν αυτή η τμηματοποίηση του αρχικού live streaming υλικού. Επίσης, από την πλευρά του χρήστη το σύστημα απαιτούσε να εγκατασταθούν κάποια προγράμματα έτσι ώστε ο χρήστης να έχει δυνατότητα να βλέπει όλα τα γεγονότα που είχε δημιουργήσει ο broadcaster.

Τέλος, το σύστημα του LiveTVSimulator δεν μπορεί να υποστηρίξει ότι διαθέτει τα πλεονεκτήματα που παρουσιάζει η δομημένη προσέγγιση μιας και δεν υπάρχει κάποιο εργαλείο που να διαχειρίζεται τα δεδομένα. Δηλαδή, το μοντέλο δεν είναι πλήρως παραμετροποιημένο για να μπορεί να διαχειριστεί χωρίς πολλές αλλαγές διαφορετικές πολυμεσικές παρουσιάσεις. Για παράδειγμα, οι συγγραφείς δεν παρουσίασαν κάποιο εργαλείο που θα επέτρεπε στον broadcaster να προσθέσει καινούργια διαφορετικά γεγονότα. Επίσης, επειδή όλη η πληροφορία βρίσκεται μέσα στο πολυμεσικό αντικείμενο



δεν είναι επεξεργάσιμη και εύκολα προσβάσιμη. Δηλαδή, ο χρήστης δεν μπορεί να αναζητήσει εύκολα την πληροφορία μέσα στο διαδίκτυο και να προβάλλει ξανά τα highlights μετά τη λήξη της ζωντανής μετάδοσης.

Στο μοντέλο που θα παρουσιάσουμε, θα προσπαθήσουμε να δώσουμε λύση στο πρόβλημα του Late tuning, χρησιμοποιώντας ως βάση για το συγχρονισμό το πρότυπο SMIL. Σκοπός μας είναι να δημιουργήσουμε μια πλήρως παραμετροποιημένη εφαρμογή για τον broadcaster που θα του επιτρέπει να ορίζει και να δημιουργεί γεγονότα για οποιαδήποτε ζωντανή, πολυμεσική παρουσίαση. Θα χρησιμοποιήσουμε τα πρωτόκολλα επικοινωνίας του διαδικτύου που επιτρέπουν τη μετάδοση αντικειμένων μέσω του διαδικτύου για να μπορεί ο viewer κατόπιν να προβάλει αυτά τα γεγονότα. Μεταξύ του πομπού και του δέκτη θα υπάρχει ενδιάμεσα μια βάση δεδομένων που θα αποθηκεύει την πληροφορία, καθιστώντας την αντικείμενο αναζήτησης αλλά και προσβάσιμη στο μέλλον από οποιοδήποτε χρήστη μέσα από το διαδίκτυο.

## 4. ΥΛΟΠΟΙΗΣΗ ΜΟΝΤΕΛΟΥ

Το μοντέλο που θα υλοποιήσουμε θα εφαρμοστεί σε ζωντανές, διαδραστικές πολυμεσικές εφαρμογές που παρουσιάζονται στο διαδίκτυο που στο εξής για λόγους απλότητας θα αναφέρονται ως θεάματα (shows). Συγκεκριμένα, θέλουμε το σύστημά μας να μπορεί να δημιουργεί πολυμεσικό περιεχόμενο δυναμικά (live generated context) αλλά και να δίνει λύση στο πρόβλημα του late tuning. Για να επεξηγήσουμε καλύτερα τον σκοπό της υλοποίησης του μοντέλου θα ξεκινήσουμε παρουσιάζοντας μερικά χαρακτηριστικά παραδείγματα στα οποία το μοντέλο μας θα πρέπει να μπορεί να εφαρμοστεί επιτυχώς.

Μια από τις πιο συνηθισμένες ζωντανές πολυμεσικές παρουσιάσεις στο σύγχρονο κόσμο είναι η μετάδοση ενός αθλητικού γεγονότος. Η μετάδοση του θεάματος είναι ζωντανή και μπορεί να γίνει είτε μέσω του διαδικτύου (live streaming) είτε μέσω της ψηφιακής τηλεόρασης και χαρακτηρίζεται ως πολυμεσική γιατί περιέχει βίντεο και ήχο. Άλλες παρόμοιες παρουσιάσεις που εμπίπτουν στο γενικό πλαίσιο του ενδιαφέροντός μας είναι η μετάδοση μια μουσικής συναυλίας, η απονομή των βραβείων Oscar ή μια ζωντανή παρουσίαση ενός νέου προϊόντος.

Ο σκοπός του μοντέλου μας είναι η παρουσίαση να γίνει διαδραστική, δηλαδή ο δημιουργός της να μπορεί να αλληλεπιδράσει με το σύστημα για να εμφανίσει στον θεατή είτε επιπλέον, γνωστό εκ των προτέρων, περιεχόμενο είτε να δημιουργήσει περιεχόμενο δυναμικά. Για παράδειγμα στην πρώτη περίπτωση, κατά την διάρκεια της απονομής των Oscar, ο θεατής θα μπορεί να προβάλει πληροφορίες για κάποιον ηθοποιό, όπως σε ποιες ταινίες έχει παίξει, πόσες φορές έχει βρεθεί υποψήφιος κ.α.

Το κύριο όμως ζητούμενο του συστήματός μας είναι η δεύτερη περίπτωση όπου δημιουργείται δυναμικό περιεχόμενο για γεγονότα που ξέρουμε ότι μπορεί να συμβούν αλλά δεν ξέρουμε ακριβώς πότε. Αυτή η αλληλεπιδραστική προσέγγιση θα μπορέσει να δώσει απάντηση και στο πρόβλημα του late tuning όταν ένας θεατής δεν συντονίζεται από την αρχή μιας ζωντανής μετάδοσης και χάνει ένα σημαντικό γεγονός (event), όπως την επίτευξη ενός γκολ σε έναν ποδοσφαιρικό αγώνα ή την εκκίνηση σε ένα αγώνα φόρμουλα1. Το σύστημά μας θα πρέπει να δίνει στον χρήστη τη δυνατότητα να προβάλλει αυτά τα events τα οποία δεν είδε σε real-time μετάδοση.

#### 4.1 ΘΕΩΡΗΤΙΚΗ ΕΠΕΞΗΓΗΣΗ ΜΟΝΤΕΛΟΥ

Το μοντέλο που θα υλοποιηθεί θα αφορά την δημιουργία ενός ολοκληρωμένου συστήματος συγγραφής και διαχείρισης SMIL αρχείων που θα λύνουν το πρόβλημα του Late Tuning σε μια Live streaming πολυμεσική παρουσίαση. Συγκεκριμένα, το μοντέλο που θα περιγράψουμε θα καθορίζει τις σχέσεις συγχρονισμού που θα βασίζονται σε γεγονότα που δεν είναι εκ των προτέρων χρονικά καθορισμένα αλλά συμβαίνουν μέσα στην παρουσίαση.

Το μοντέλο θα επικεντρώνεται στη σωστή διαχείριση των extra δεδομένων που θα βασίζονται σε γεγονότα (events) που ενδεχομένως να συμβούν. Λόγω της φύσης της πολυμεσικής παρουσίας ασχολούμαστε με το ποια γεγονότα μπορεί να συμβούν κατά την διάρκειά της και όχι πότε αυτά θα συμβούν. Το μοντέλο θα είναι γενικό και πλήρως παραμετροποιημένο και θα μπορεί να εφαρμοστεί σε οποιαδήποτε πολυμεσική παρουσίαση, μιας και ο broadcaster θα μπορεί να δημιουργήσει τις δικές του παρουσιάσεις (show) αλλά και τα δικά του γεγονότα που αφορούν την παρουσίαση.

Τα διαφορετικά πολυμεσικά γεγονότα αντιπροσωπεύονται από αρχεία SMIL ενώ οι σχέσεις μεταξύ τους ορίζονται από δεδομένα που καταγράφονται σε μια βάση δεδομένων. Θα ασχοληθούμε τόσο με την πλευρά του πομπού (broadcaster) όσο και του θεατή/δέκτη (viewer/receiver) και ποιες θα πρέπει να είναι οι ενέργειές τους ώστε να διαχειριστούν τα γεγονότα.

Η εφαρμογή μας θα στηριχτεί στις παρακάτω τεχνολογίες:

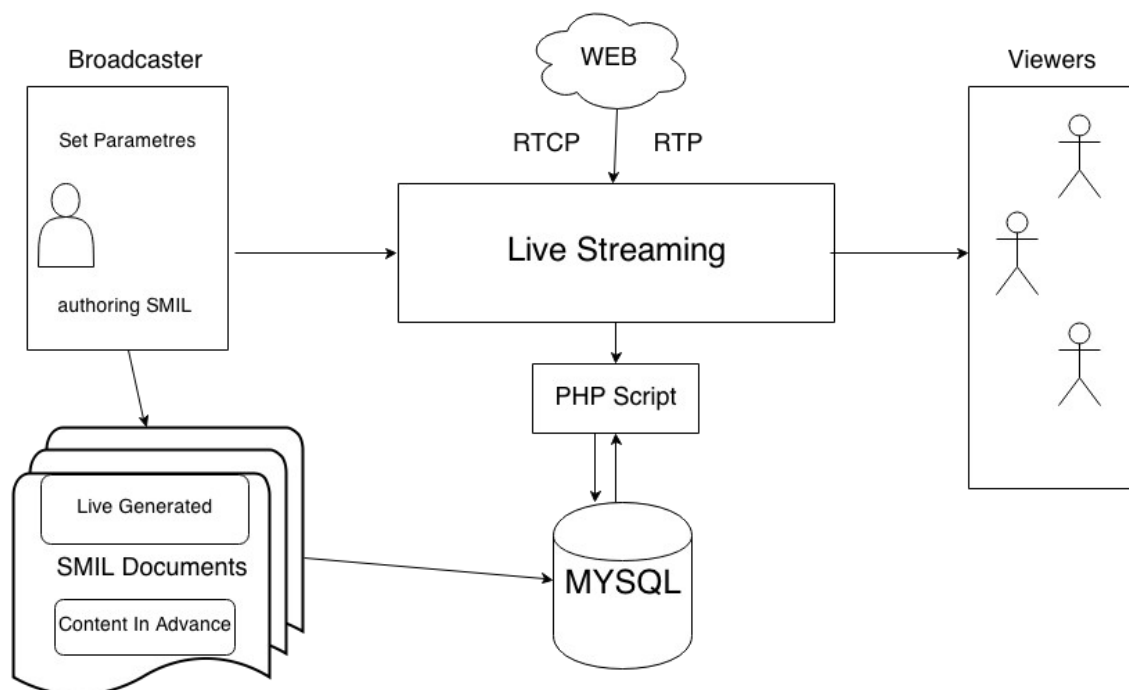
1. **PHP.** Χρησιμοποιούμε την PHP έκδοση 5.3.0 για την αλληλεπίδραση του broadcaster με το σύστημα, για τη σύνδεση με τη βάση δεδομένων, για την δημιουργία των SMIL αρχείων αλλά και για να παρουσιάζονται σωστά οι φόρμες στις διαδικτυακές σελίδες σε όλους τους browsers.
2. **SMIL.** Χρησιμοποιούμε τη δομημένη markup γλώσσα SMIL για να διαχειριστούμε τα θέματα συγχρονισμού στην πολυμεσική παρουσίαση.
3. **MySQL.** Χρησιμοποιούμε τη βάση δεδομένων έκδοση MySQL 5.1.36 για να ξεχωρίσουμε το περιεχόμενο από την παρουσίαση και να εξασφαλίσουμε τα χαρακτηριστικά accessibility και reusability. Επιλέξαμε τη τεχνολογία MySQL γιατί

χαρκτηρίζεται ως μια γρήγορη, αξιόπιστη σχεσιακή βάση δεδομένων ανοικτού κώδικα που διανέμεται δωρεάν και μπορεί να συνεργαστεί με πολλές πλατφόρμες, προσφέροντας ανεξαρτησία στις εφαρμογές.

4. **Xampp.** Χρησιμοποιούμε ως http εξυπηρετητή στο σύστημά μας την τεχνολογία Xampp έκδοση. 3.2.1.

Τα πρωτόκολλα που θα χρησιμοποιηθούν για τις πολυμεσικές παρουσιάσεις με τη βοήθεια της γλώσσας SMIL είναι το κλασσικά διαδικτυακό πρωτόκολλο HTTP αλλά και το πρωτόκολλο RTP (Real-time Transport Protocol) και RTCP (RTP Control Protocol). Τα τελευταία πρωτόκολλα θα χρησιμοποιηθούν για την live μετάδοση των παρουσιάσεων. Τα δύο αυτά πρωτόκολλα δεν εγγυώνται τη παράδοση των δεδομένων σε πραγματικό χρόνο (real-time) αλλά προσφέρουν ένα πλαίσιο μέσα από το οποίο μια εφαρμογή μπορεί να υλοποιήσει μια τέτοια ακριβή παράδοση.

Η αρχιτεκτονική του συστήματος φαίνεται στο παρακάτω σχεδιάγραμμα:

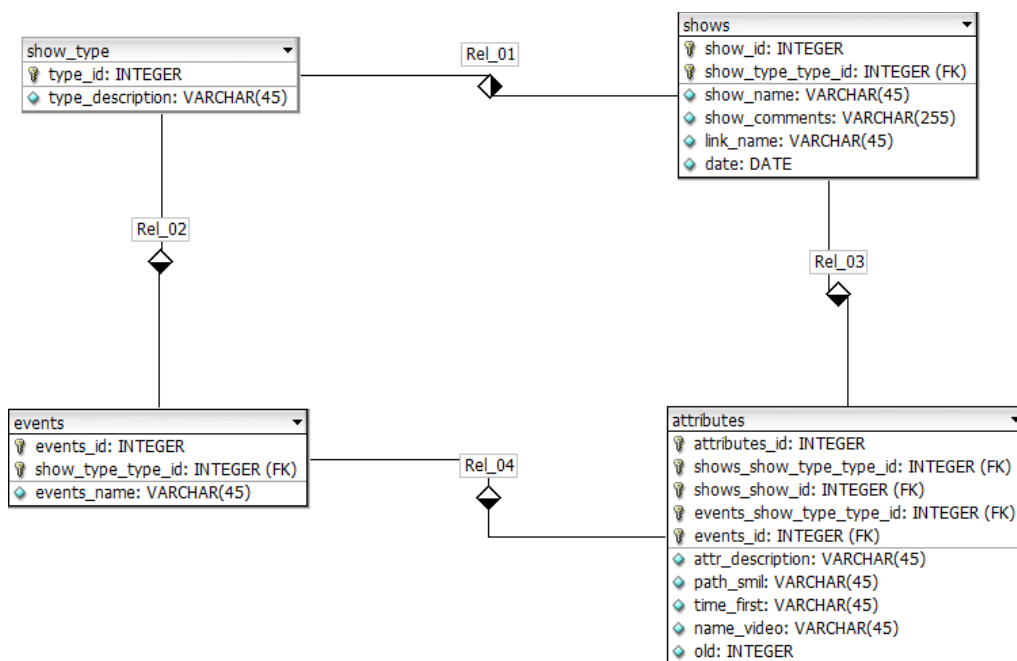


Όπως φαίνεται και από το σχεδιάγραμμα, το σύστημα θα παρέχει εργαλεία για την δημιουργία και παρουσίαση πολυμέσων μέσω της βοήθειας των SMIL αρχείων. Θα παρέχει ένα φιλικό περιβάλλον για τον broadcaster ως προς την αναζήτηση δημιουργία των αρχείων και ένα αντίστοιχο για τον χρήστη για την παρουσίαση των ίδιων αρχείων.

Τα δεδομένα που επεξεργάζεται στο σύστημα και θα εισαχθούν στην βάση δεδομένων μας έχουν δομηθεί πάνω στην αντικειμενοστραφή λογική. Έτσι το μοντέλο δεδομένων μας περιέχει:

- Κλάσεις Ci που αφορούν τις κατηγορίες θεαμάτων (show types). Παραδείγματα τέτοιων κλάσεων μπορεί να είναι η κατηγορία *ποδοσφαιρικό αγώνας*, η κατηγορία *μουσική συναυλία* κτλ
- Λίστα Li που αφορά τα γεγονότα (event) που μπορεί να έχει κάθε κλάση Ci. Για παράδειγμα, η κλάση *ποδοσφαιρικό αγώνας* μπορεί να έχει λίστα γεγονότων: {γκολ, πέναλτι, κόκκινη κάρτα}
- Αντικείμενα Oi που αφορούν ένα συγκεκριμένο θέαμα (show) και πρόκειται για Instances κάποιας ορισμένης κλάσης Ci. Για παράδειγμα, ο αγώνας *Παναθηναϊκός – ΠΑΟΚ* είναι αντικείμενο της κλάσης *ποδοσφαιρικό αγώνας*.
- Ιδιότητες/Γνωρίσματα (attributes) που αφορούν το συγκεκριμένο αντικείμενο Oi. Για παράδειγμα, για το αντικείμενο *Παναθηναϊκός – ΠΑΟΚ* γνωρίσματα μπορεί να είναι τα ακόλουθα {γκολ, 1-0, 12/03/2014, 20 λεπτό}

Με βάση αυτή τη λογική δημιουργήθηκε και η βάση δεδομένων, της οποίας το UML διάγραμμα παρουσιάζεται παρακάτω:



## 4.2 BROADCASTER

Ο broadcaster είναι ουσιαστικά ο δημιουργός της πολυμεσικής παρουσίασης αλλά και ο υπεύθυνος διαχείρισής της. Είναι, δηλαδή, αυτός που θα δώσει αρχικά τις αναγκαίες πληροφορίες για την παρουσίαση, όπως τον τύπο του θεάματος που πρόκειται να μεταδοθεί ζωντανά. Ακολούθως, είναι αυτός που θα “δημιουργήσει” τα replay, ορίζοντας τις κατάλληλες παραμέτρους και θα τα προωθήσει για προβολή στον θεατή. Το σύστημα που προτείνουμε είναι πλήρως παραμετροποιημένο ώστε ο broadcaster να μπορεί να εισάγει νέους τύπους θεαμάτων αλλά και να μπορεί να ορίσει νέα γεγονότα για τα οποία θέλει να δημιουργήσει replay.

Στη συνέχεια θα παρουσιάσουμε τις απαραίτητες διαδικασίες που πρέπει να ακολουθήσει ο broadcaster αλλά και τις δυνατότητες αλληλεπίδρασης που του προσφέρει το σύστημά μας ώστε να αλλάξει τη ροή της παρουσίασης.

### Αρχική σελίδα (broadcaster.php)

Στην αρχική σελίδα ο broadcaster είναι υπεύθυνος να ορίσει τις αναγκαίες παραμέτρους ώστε να ξεκινήσει η ζωντανή μετάδοση της πολυμεσικής παρουσίασης (θέαμα). Ουσιαστικά, ο broadcaster πρέπει να συμπληρώσει τις παρακάτω απαραίτητες πληροφορίες:

- **Κατηγορία Θεάματος.** Ο broadcaster πρέπει να επιλέξει μια από τις υπάρχουσες κατηγορίες θεάματος, π.χ. ποδοσφαιρικό αγώνα, απονομή Oscar, μουσική συναυλία κτλ. Αυτή η πληροφορία είναι απαραίτητη γιατί ανάλογα με την κατηγορία θεάματος, ο broadcaster θα έχει στην διάθεσή του τα αντίστοιχα event (π.χ. γκολ, κόκκινη κάρτα, απονομή 1<sup>ου</sup> γυναικείου ρόλου κτλ) για να δημιουργήσει replay.
- **Περιγραφή – Σχόλια.** Ο broadcaster δίνει μια σύντομη περιγραφή του συγκεκριμένου θεάματος που πρόκειται να μεταδοθεί ζωντανά. Για παράδειγμα στην κατηγορία «μουσική συναυλία», ένα θέαμα θα μπορούσε να ήταν «Scorpions, Αθήνα».
- **Link Live Streaming.** Ο broadcaster συμπληρώνει το link για τη Live streaming παρουσίαση. Η ύπαρξη αυτής της πληροφορίας αν και θα μπορούσε να παραληφθεί, θα βοηθήσει στην διαχείριση των πολυμεσικών αντικειμένων μέσω της γλώσσας SMIL.

Το σύστημα ελέγχει ότι ο broadcaster έχει συμπληρώσει σωστά όλες τις πληροφορίες και μάλιστα ότι αυτό το θέαμα δεν έχει εισαχθεί ξανά στη βάση δεδομένων του συστήματός μας. Πιο συγκεκριμένα, δεν μπορεί να εισαχθεί θέαμα ίδιας κατηγορίας, με την ίδια περιγραφή και την ίδια ημερομηνία.

11-03-2014	
Υπάρχει event με το ίδιο όνομα ή τα στοιχεία δεν είναι συμπληρωμένα όλα σωστά.	
Κατηγορία Θεάματος:	-Επιλέξτε-
Περιγραφή-Σχόλια:	<input type="text"/>
Link live streaming:	<input type="text"/> <input type="button" value="Συνέχεια"/>
+ Προσθήκη νέας Κατηγορίας Θεάματος + Προσθήκη νέου event σε κατηγορία θεάματος	

Εκτός από την δυνατότητα προσθήκης νέου θεάματος, ο broadcaster έχει την δυνατότητα στην αρχική σελίδα να εισάγει επίσης νέα κατηγορία θεάματος αλλά και νέο event σε κατηγορία θεάματος

### Προσθήκη νέας κατηγορίας θεάματος

11-03-2014	
Κατηγορία Θεάματος:	-Επιλέξτε-
Περιγραφή-Σχόλια:	<input type="text"/>
Link live streaming:	<input type="text"/> <input type="button" value="Συνέχεια"/>
+ Προσθήκη νέας Κατηγορίας Θεάματος + Προσθήκη νέου event σε κατηγορία θεάματος	
Περιγραφή :	<input type="text"/> <input type="button" value="Προσθήκη"/>

Ο broadcaster μπορεί να δει τη λίστα με τις διαθέσιμες κατηγορίες θεάματος (π.χ. ποδοσφαιρικός αγώνας). Αν το θέαμα που πρόκειται να μεταδοθεί ζωντανά δεν ανήκει σε καμία από τις ήδη υπάρχουσες κατηγορίες θεάματος τότε μπορεί εύκολα να προστεθεί μία νέα. Η λίστα με τις διαθέσιμες κατηγορίες θεάματος ανανεώνεται αυτόματα και ο broadcaster μπορεί να συνεχίσει με την διαχείριση του συγκεκριμένου θεάματος που πρόκειται να προβληθεί.

### Προσθήκη νέου event σε κατηγορία θεάματος

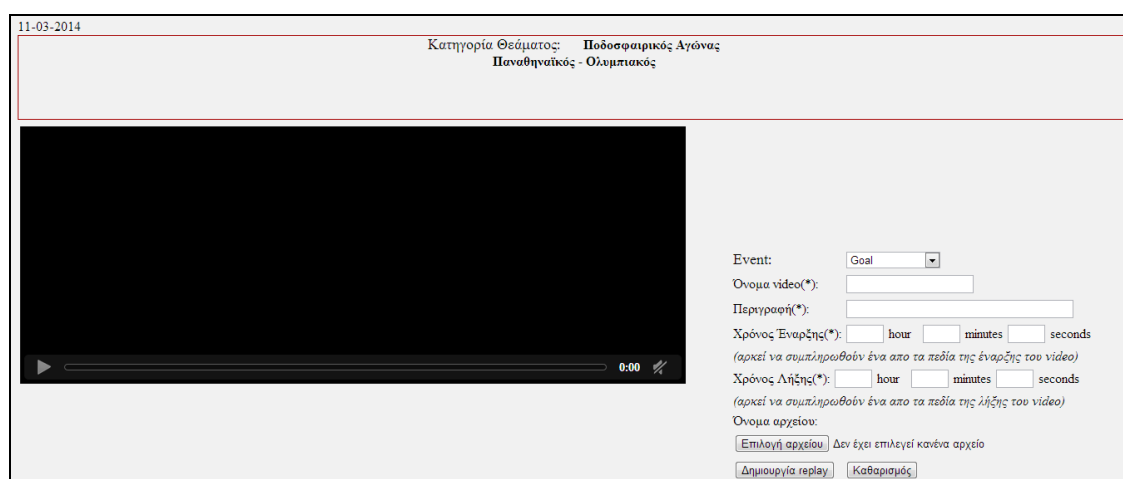
11-03-2014	
Κατηγορία Θεάματος:	-Επιλέξτε-
Περιγραφή-Σχόλια:	<input type="text"/>
Link live streaming:	<input type="text"/> <input type="button" value="Συνέχεια"/>
+ Προσθήκη νέας Κατηγορίας Θεάματος + Προσθήκη νέου event σε κατηγορία θεάματος	
Κατηγορία Θεάματος:	-Επιλέξτε-
Περιγραφή :	<input type="text"/> <input type="button" value="Προσθήκη"/>

Από την αρχική σελίδα ο broadcaster μπορεί να διαχειριστεί και τα event που αφορούν τις κατηγορίες θεάματος. Πιο συγκεκριμένα, ο broadcaster μπορεί να επιλέξει μια από τις διαθέσιμες κατηγορίες θεάματος και να προσθέσει event για αυτήν την επιλεγμένη κατηγορία θεάματος. Για παράδειγμα

μπορεί να επιλέξει τον «ποδοσφαιρικό αγώνα» ως κατηγορία θεάματος και να προσθέσει το «Πέναλτι» ως νέο event.

### Κεντρική σελίδα broadcaster (set\_parameters.php)

Όταν ο broadcaster επιλέξει το συγκεκριμένο θέαμα που θα μεταδοθεί ζωντανά και πατήσει το κουμπί *Συνέχεια* τότε μεταφέρεται στην κεντρική σελίδα που φαίνεται παρακάτω. Στην κορυφή της οθόνης φαίνεται η κατηγορία θεάματος που επιλέχθηκε αλλά και η περιγραφή του θεάματος που συμπληρώθηκε στην αρχική σελίδα.



11-03-2014

Κατηγορία Θεάματος: Ποδοσφαιρικός Αγώνας  
Πανσθηναϊκός - Ολυμπιακός

Event: Goal

Όνομα video(\*):

Περιγραφή(\*):

Χρόνος Έναρξης(\*): hour minutes seconds  
(αρκεί να συμπληρωθούν ένα από τα πεδία της έναρξης του video)

Χρόνος Λήξης(\*): hour minutes seconds  
(αρκεί να συμπληρωθούν ένα από τα πεδία της λήξης του video)

Όνομα αρχείου:

Δεν έχει επιλεγεί κανένα αρχείο

Σε αυτήν την σελίδα, ο broadcaster μπορεί να παρακολουθήσει τη live streaming μετάδοση (στο αριστερό κομμάτι της οθόνης) αλλά μπορεί να διαχειριστεί και την δημιουργία και προβολή των replay. Οι βασικές δυνατότητες που παρέχονται στον χρήστη είναι η δυνατότητα της δημιουργίας replay αλλά και της αναζήτησης παλιών replay.

Αρχικά, ο broadcaster, παρακολουθώντας τη ζωντανή μετάδοση μπορεί να θελήσει να δημιουργήσει ένα νέο replay που θα μπορούν μετά οι χρήστες να προβάλλουν (live generated content). Η φόρμα που πρέπει να συμπληρώσει ο broadcaster φαίνεται παραπάνω στο δεξί κομμάτι της οθόνης. Έτσι, ο broadcaster πρέπει να συμπληρώσει τα εξής στοιχεία:

- **Event.** Ο χρήστης broadcaster πρέπει να επιλέξει το event το οποίο αφορά το replay το οποίο θα δημιουργηθεί. Οι επιλογές που εμφανίζονται στη λίστα των event περιορίζονται σε αυτές που αφορούν τη συγκεκριμένη κατηγορία θεάματος που επιλέχθηκε στην αρχική σελίδα.
- **Όνομα video.** Πρόκειται για το όνομα του replay όπως θα καταχωρηθεί στην βάση δεδομένων. Είναι υποχρεωτικό πεδίο που πρέπει να συμπληρωθεί.

- **Περιγραφή.** Πρόκειται για την περιγραφή του replay όπως θα καταχωρηθεί στην βάση δεδομένων. Είναι υποχρεωτικό πεδίο που πρέπει να συμπληρωθεί και αποτελεί αντικείμενο αναζήτησης (keywords).
- **Χρόνος Έναρξης και Χρόνος Λήξης.** Ουσιαστικά, ο broadcaster ορίζει την έναρξη και λήξη του replay video. Είναι υποχρεωτικά πεδία και εκφράζονται σε ώρες, λεπτά και δευτερόλεπτα από την έναρξη της live streaming μετάδοσης.
- **Επιλογή αρχείου.** Πρόκειται για την προσθήκη περιεχομένου γνωστού εκ των προτέρων (extra content) στο video replay, το οποίο αναπαράγεται σε σειρά μετά το video. Η επιλογή αυτή θα εξηγηθεί λεπτομερέστερα αργότερα.

Συμπληρώνοντας τις απαραίτητες πληροφορίες και πατώντας το κουμπί *Δημιουργία replay* τότε δημιουργούνται:

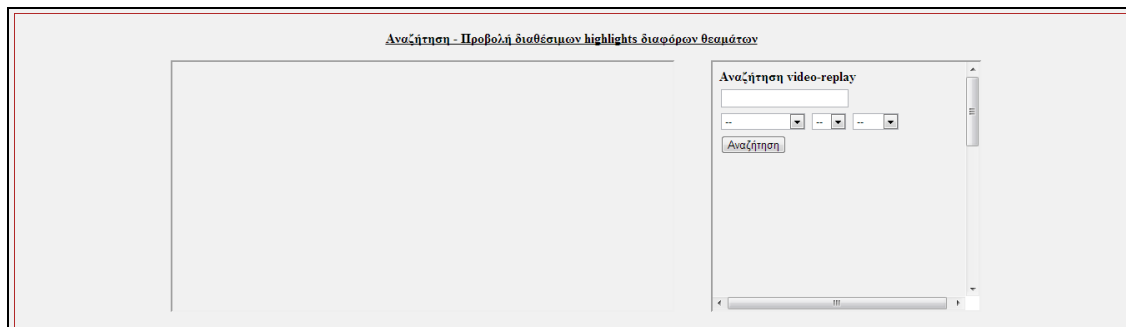
1. Ένα αρχείο κειμένου που περιέχει το όνομα του θεάματος που συμπληρώθηκε στην αρχική σελίδα αλλά και η περιγραφή που δόθηκε για το συγκεκριμένο event.
2. Ένα αρχείο SMIL που περιέχει όλες τις απαραίτητες πληροφορίες για να αναπαραχθεί το replay. Επίσης, στο SMIL αρχείο δημιουργείται link στο αρχείο κειμένου που δημιουργήθηκε στο προηγούμενο βήμα. Έτσι, όταν ο θεατής αναπαράγει το replay video θα έχει πάντα και την πληροφορία για το ποιο θέαμα αφορά.
3. Μια εγγραφή στην βάση που περιέχει την πληροφορία για το SMIL αρχείο.

Επιπλέον, ο broadcaster έχει την δυνατότητα να προσθέσει και ένα επιπλέον βίντεο (extra content) στο replay το οποίο θα αναπαραχθεί μόλις τελειώσει το βίντεο του replay (sequential). Το επιπλέον αυτό βίντεο υπάρχει ήδη καταχωρημένο στον server (content in advance) και δεν δημιουργείται δυναμικά. Με αυτόν τρόπο μπορεί εύκολα να προστεθεί ο πανηγυρισμός από μια κερκίδα ή ο ύμνος μιας ομάδας ακόμα και μια διαφήμιση χορηγού.

Με την δημιουργία ενός replay, δημιουργείται ένα SMIL αρχείο που περιέχει όλη την πληροφορία που αφορά το κομμάτι του live streaming που πρέπει να αποθηκευτεί και να επιδειχτεί καθώς και άλλες επιπλέον πληροφορίες, όπως την ανάλυση ποιότητας κτλ. Σε δεύτερη φάση, δημιουργείται μια εγγραφή στη βάση δεδομένων που περιέχει όλες τις πληροφορίες που παρέχει ο broadcaster για το γεγονός μαζί με την πληροφορία με τη σύνδεση στο αρχείο SMIL που δημιουργήθηκε στην πρώτη φάση. Με την δημιουργία του replay από τον broadcaster, αυτόματα ενημερώνεται η λίστα που βλέπει ο θεατής όπως θα αναλύσουμε στο επόμενο κεφάλαιο.

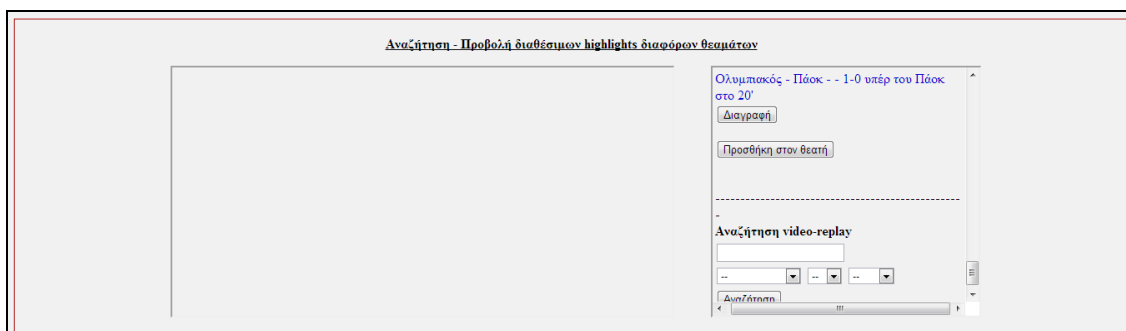


Η άλλη δυνατότητα που δίνεται στον broadcaster είναι αυτή της αναζήτησης και προσθήκης replay που έχουν δημιουργηθεί στο παρελθόν όπως φαίνεται παρακάτω:



Ο broadcaster μπορεί να αναζητήσει κάποιο παλιό video replay - ή και αυτά που έχει δημιουργήσει για τη ζωντανή μετάδοση – δίνοντας ως κριτήρια keywords ή/και ημερομηνία δημιουργίας του video(provoli\_smil\_files.php). Τα keywords αναζητούνται στην περιγραφή του θεάματος, στο όνομα του video replay αλλά και στην περιγραφή του video replay.

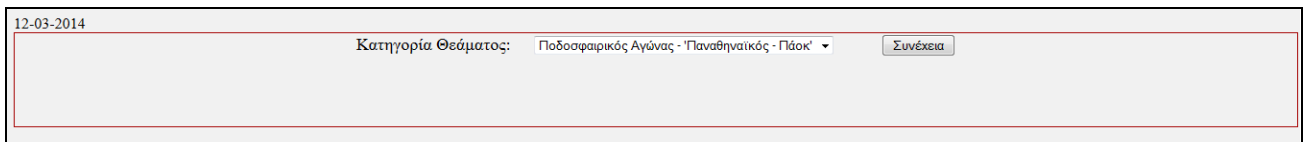
Εφόσον βρεθούν video replay κατά την αναζήτηση, τότε ο broadcaster έχει τις επιλογές που φαίνονται παρακάτω:



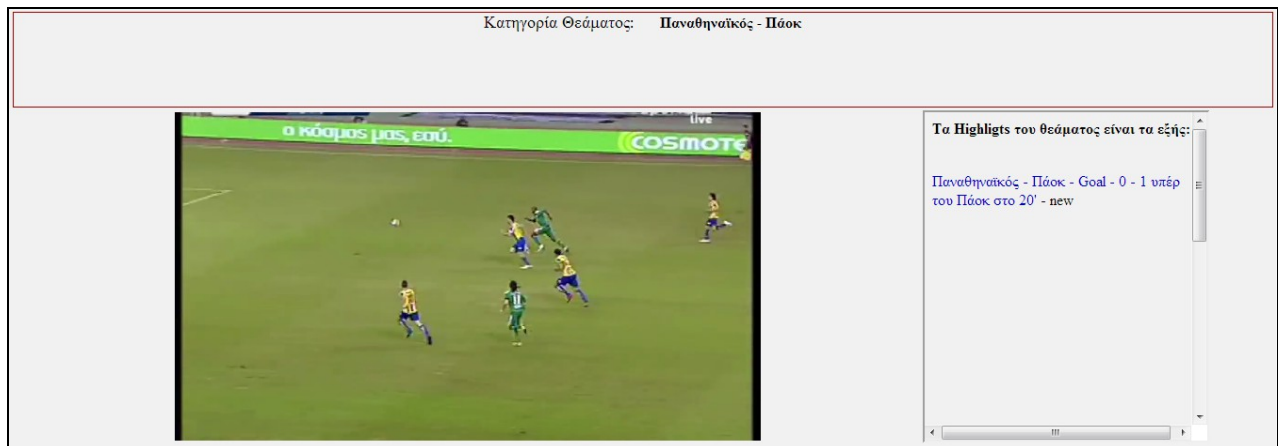
- **Διαγραφή.** Ο broadcaster μπορεί να διαγράψει κάποιο replay. Η διαγραφή αφορά μόνο τα replay που δημιουργήθηκαν για τη συγκεκριμένη ζωντανή μετάδοση. Η διαδικασία αυτή περιλαμβάνει τόσο την διαγραφή της εγγραφής στην βάση, όσο και την διαγραφή του αντίστοιχου SMIL αρχείου.
- **Προσθήκη στον θεατή.** Το video replay που αναζητήθηκε μπορεί να προστεθεί στη λίστα των διαθέσιμων replay που βλέπει ο θεατής.
- **Προβολή.** Ο broadcaster μπορεί να προβάλλει τα video που αναζήτησε ώστε να βεβαιωθεί ότι πρόκειται για το replay που έψαχνε πριν το προσθέσει στη λίστα του θεατή. Η προβολή αυτών των video γίνεται στο αριστερό πλαίσιο της παραπάνω φωτογραφίας.

### 4.3 VIEWER

Ο θεατής (viewer) είναι ο χρήστης που παρακολουθεί το live streaming, δηλαδή τη ζωντανή πολυμεσική παρουσίαση. Μόλις συνδεθεί στο σύστημα, ο θεατής επιλέγει το θέαμα που θέλει να παρακολουθήσει από μια λίστα με τις διαθέσιμες παρουσιάσεις που υπάρχουν στην συγκεκριμένη ημερομηνία (την ημερομηνία εισαγωγής στο σύστημα)(viewer.php).



Ο θεατής βλέπει την παρακάτω οθόνη (viewer.php) :



Στην κορυφή της οθόνης φαίνεται η πληροφορία για το θέαμα που θα μεταδοθεί ζωντανά ενώ στο αριστερό κομμάτι της οθόνης εμφανίζεται η live streaming μετάδοση. Στο δεξί μέρος της οθόνης φαίνονται τα highlight που αφορούν το συγκεκριμένο θέαμα(pronoli\_viewer.php). Όπως αναφέραμε σε αυτή τη λίστα υπάρχουν τα replays που έχει δημιουργήσει ο broadcaster και αφορούν την συγκεκριμένη μετάδοση (live generated content). Μάλιστα, ο θεατής μπορεί να δει ποια είναι η τελευταία προσθήκη highlight που έχει γίνει από τον broadcaster και χαρακτηρίζεται από τη λέξη *new*. Ο θεατής έχει την δυνατότητα στην ίδια σελίδα να προβάλλει αυτά τα highlight, δίνοντας λύση στο πρόβλημα του late tuning.

Επίσης, υπάρχει μια λίστα κάτω από την επικεφαλίδα *Δείτε Επίσης* που περιέχει τα παλιά video που έχουν συσχετιστεί με αυτήν την μετάδοση(pronoli\_old.php). Ο broadcaster έχει την επιλογή να αναζητήσει παλιά σχετικά video και να την προσθέσει στις επιλογές του θεατή. Ο θεατής μπορεί φυσικά να προβάλλει και αυτά τα video.



Τέλος, ο θεατής έχει την δυνατότητα να προβάλλει κάποια video που το περιεχόμενο είναι εκ των προτέρων γνωστό και σχετικό με το θέαμα που θα παρακολουθήσει (content in advance). Για παράδειγμα, σε έναν ποδοσφαιρικό αγώνα ο θεατής θα μπορεί να δει video με πληροφορίες όπως την βαθμολογία, το πλήρες πρόγραμμα της αγωνιστικής, στατιστικά για γκολ κτλ.

#### 4.4 ΠΕΙΡΑΜΑΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ

Στην παρούσα ενότητα θα αναφερθούμε στην πειραματική αξιολόγηση του συστήματός μας. Αρχικά, θα δούμε ένα σενάριο ζωντανής μετάδοσης ποδοσφαιρικού αγώνα και θα εξετάσουμε πως ανταποκρίθηκε το σύστημά μας. Στη συνέχεια θα αξιολογήσουμε το σύστημά μας με βάση τα κριτήρια που έχουμε ήδη θέσει και τέλος θα προτείνουμε ιδέες που μπορούν να υλοποιηθούν στο μέλλον για να βελτιώσουν το μοντέλο μας.

#### ΣΕΝΑΡΙΟ

Θεωρούμε ότι θα μεταδοθεί ζωντανά ένας ποδοσφαιρικός αγώνας που ξεκινάει στις 12/03/2014 ώρα 21:45. Η μετάδοση ενός τέτοιου θεάματος γίνεται για πρώτη φορά οπότε πρέπει να εισαχθεί καινούργια κατηγορία. Επίσης, εφόσον η κατηγορία πρέπει να δημιουργηθεί, θα πρέπει να δημιουργηθούν και τα αντίστοιχα event. Ως πιθανά event θα εισάγουμε την επίτευξη ενός γκολ, την εκτέλεση ενός πέναλτι και την κόκκινη κάρτα.

#### 21:30

Ο broadcaster μπαίνει στο σύστημά μας και αφού ελέγξει ότι η κατηγορία θεάματος *ποδοσφαιρικός αγώνας* δεν υπάρχει, την εισάγει.

12-03-2014

Κατηγορία Θεάματος: -Επιλέξτε-

Περιγραφή-Σχόλια:

Link live streaming:  Συνέχεια

+ Προσθήκη νέας Κατηγορίας Θεάματος

Περιγραφή: Ποδοσφαιρικός Αγώνας Προσθήκη

+ Προσθήκη νέου event σε κατηγορία θεάματος

Επίσης, δημιουργεί και τα αντίστοιχα event: γκολ, πέναλτι και κόκκινη

12-03-2014

Κατηγορία Θεάματος: -Επιλέξτε-

Περιγραφή-Σχόλια:

Link live streaming:  Συνέχεια

+ Προσθήκη νέας Κατηγορίας Θεάματος

+ Προσθήκη νέου event σε κατηγορία θεάματος

Κατηγορία Θεάματος: Ποδοσφαιρικός Αγώνας

Περιγραφή: Goal Προσθήκη

## 21:32

Ο broadcaster συμπληρώνει τα στοιχεία της κατηγορίας θεάματος, την περιγραφή του συγκεκριμένου θεάματος και το Link για το Live streaming και πατάει *Συνέχεια*.

12-03-2014

Κατηγορία Θεάματος: Ποδοσφαιρικός Αγώνας

Περιγραφή-Σχόλια: Παναθηναϊκός - Πάοκ

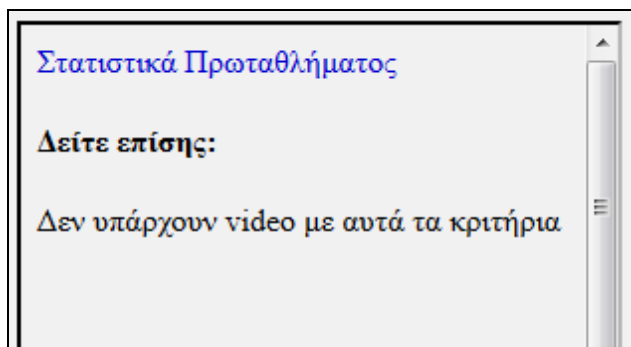
Link live streaming: rtsp://www.panathinaikospaok.gr Συνέχεια

+ Προσθήκη νέας Κατηγορίας Θεάματος

+ Προσθήκη νέου event σε κατηγορία θεάματος

## 21:45

Η μετάδοση του ποδοσφαιρικού αγώνα ξεκινά. Ο broadcaster δεν έχει δημιουργήσει ακόμα highlight. Στον viewer υπάρχει ένα video που δείχνει στατιστικές πληροφορίες για τις κόκκινες κάρτες, τα γκολ και τα πέναλτι που έχουν σημειωθεί μέχρι στιγμής στο πρωτάθλημα.



Και ο θεατής μπορεί να επιλέξει να το προβάλει.  
Στην αρχή προβάλλεται μία διαφήμιση ενός χορηγού.



Ο θεατής πατώντας το Skip Intro θα μπορεί να παρακολουθήσει τα video.

Προβολή όλων των video - Highlights



Στατιστικά Πρωταθλήματος

Δείτε επίσης:


Δεν υπάρχουν video με αυτά τα κριτήρια

**22:04**

Κατά την διάρκεια του ποδοσφαιρικού αγώνα σημειώνεται ένα από τα συγκεκριμένα γεγονότα: επίτευξη γκολ. Ο broadcaster δημιουργεί το αντίστοιχο replay.

12-03-2014

Κατηγορία Θεάματος: Ποδοσφαιρικός Αγώνας  
 Παναθηναϊκός - Πάοκ



Event: Goal

Όνομα video(\*): πρώτο goal

Περιγραφή(\*): 0-1 υπέρ του Πάοκ στο 20'

Χρόνος Έναρξης(\*): hour 18 minutes 45 seconds  
(αρκεί να συμπληρωθούν ένα απο τα πεδία της έναρξης του video)

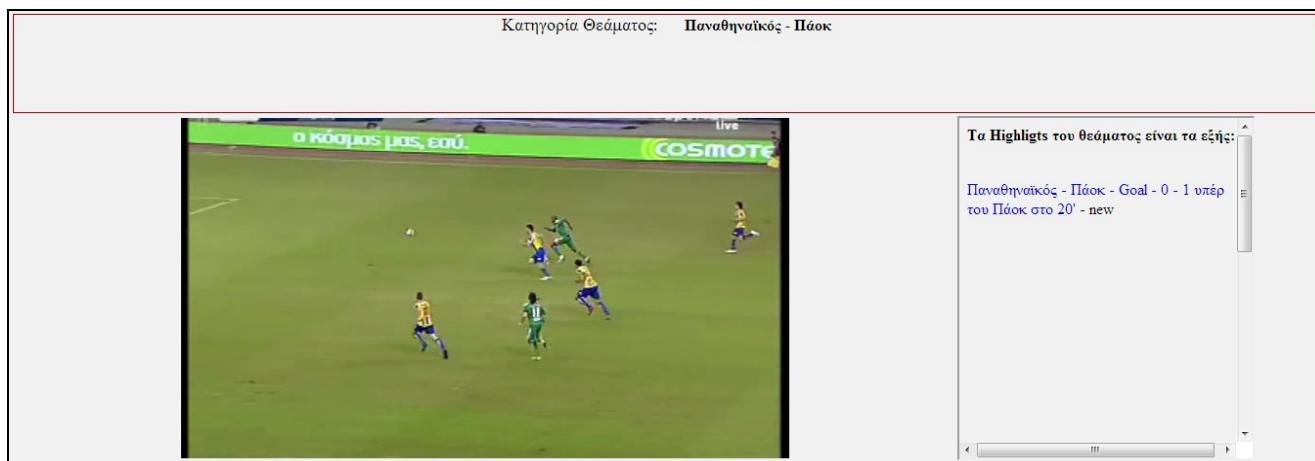
Χρόνος Λήξης(\*): hour 20 minutes 11 seconds  
(αρκεί να συμπληρωθούν ένα απο τα πεδία της λήξης του video)

Όνομα αρχείου:

Επιλογή αρχείου Κερκίδα-πάοκ.wmv

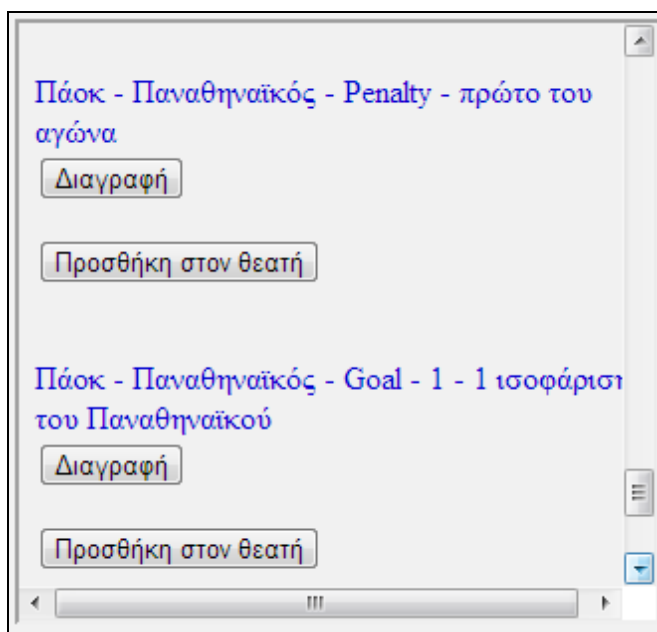
Δημιουργία replay
Καθαρισμός

Μετά από αμελητέο χρόνο, ο θεατής μπορεί να δει στη λίστα του το σχετικό replay και να το προβάλει.

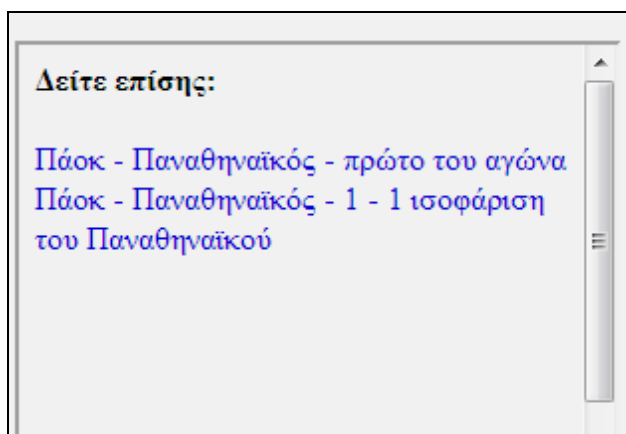


**21:58**

Ο broadcaster αναζητά άλλα παλιά video που είναι σχετικά με το event (γκολ) που μόλις σημειώθηκε. Για παράδειγμα, events από προηγούμενο αγώνα των ίδιων ομάδων.

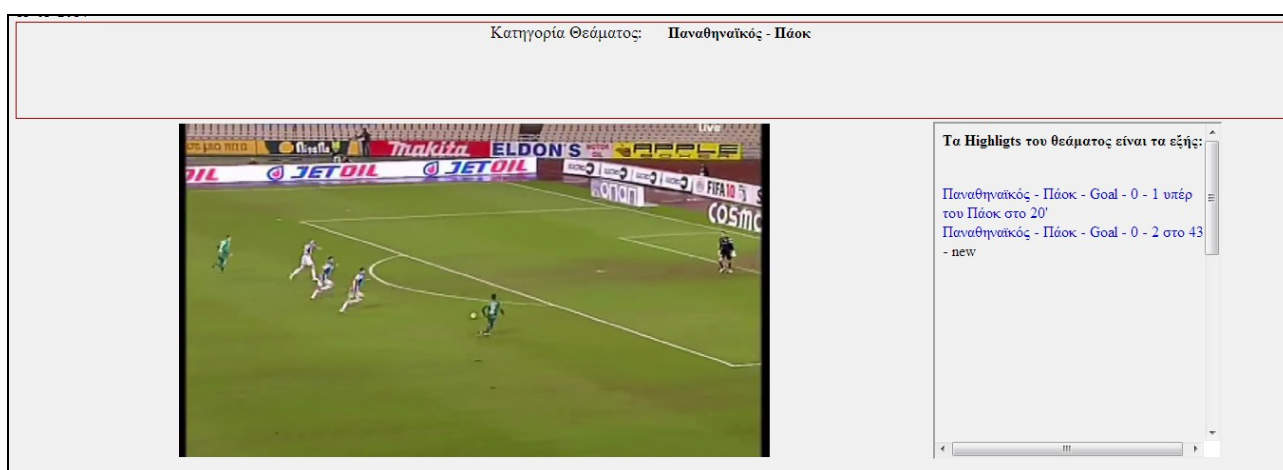


Και επιλέγει να προσθέσει στη λίστα του θεατή *Δείτε επίσης* ένα από αυτά.



**23:08**

Ένας άλλος θεατής συνδέεται ετεροχρονισμένα στη ζωντανή μετάδοση και βλέπει τη λίστα με τα γεγονότα που έχουν ήδη συμβεί.



Όσα σενάρια δοκιμάσαμε με το σύστημά μας που αφορούσαν το πρόβλημα το late tuning είχαν επιτυχές αποτέλεσμα. Έγιναν πολλές δοκιμές ώστε ο θεατής να μπορεί να προβάλλει τόσο περιεχόμενο που ήταν εκ των προτέρων γνωστό (content known in advance) όσο και περιεχόμενο που παράχθηκε δυναμικά από τον broadcaster (live generated content). Επίσης, δοκιμάστηκαν όλες οι λειτουργίες που αναφέρθηκαν στις προηγούμενες ενότητες όπως για παράδειγμα η διαγραφή ενός replay που παράχθηκε δυναμικά από τον broadcaster. Οι δοκιμές έγιναν σε Mozilla Firefox, Google Chrome και Internet Explorer 9.

Για να αξιολογήσουμε καλύτερα την αποτελεσματικότητα του μοντέλου μας θα αναλύσουμε τα χαρακτηριστικά που είχαμε αναφέρει ότι πρέπει να ικανοποιεί το σύστημά μας και είναι τα ακόλουθα:



- **Structured/Declarative.** Η χρήση της δομημένης γλώσσας SMIL μας οδήγησε σε μια γενικά δομημένη λύση που σε όλες τις δοκιμές μας βοήθησε να γνωρίζουμε τη συμπεριφορά του συστήματος.
- **Time-based.** Η υλοποίηση του συστήματος έγινε με βάση τεχνολογία που έχει ως βάση τον χρόνο με αποτέλεσμα να μην παρουσιαστούν ιδιαίτερα προβλήματα συγχρονισμού ακόμα και σε περίπλοκα σενάρια.
- **Interactivity.** Το μοντέλο μας εμπλουτίστηκε με την γλώσσα PHP για να ελέγξουμε καλύτερα την αλληλεπίδραση με τον χρήστη του συστήματος. Παράλληλα, η σύνδεση της PHP με τη βάση MYSQL μας επέτρεψε να κάνουμε το μοντέλο μας πλήρες παραμετροποιημένο. Έτσι ο broadcaster μπορούσε γρήγορα και εύκολα να θέσει τις δικές του παραμέτρους όπως να ορίσει νέα γεγονότα (event).
- **Adaptability.** Στο μοντέλο μας παρέχεται μεγάλη δυνατότητα προσαρμογής στο σύστημα που χρησιμοποιεί ο χρήστης, είτε μιλάμε για τον browser που επιλέγει είτε μιλάμε για τις υλικές δυνατότητες της συσκευής που χρησιμοποιεί (adaptability), λόγω της ιδιότητας που προσφέρει το πρότυπο SMIL να προσαρμόζει την παρουσίαση σε οποιοδήποτε περιβάλλον.
- **Accessibility.** Επειδή, το περιεχόμενο της παρουσίασης είναι αποθηκευμένο σε μια βάση δεδομένων, αυτό σημαίνει ότι ο διαδικτυακός χρήστης μπορεί να το αναζητήσει εύκολα και να το προσπελάσει οποιαδήποτε στιγμή. (accessibility). Έτσι, τα περιεχόμενα των πολυμεσικών παρουσιάσεων (SMIL αρχεία) είναι διαθέσιμα στους χρήστες και μετά την λήξη της ζωντανής μετάδοσης.
- **Reusability.** Το μοντέλο είναι πλήρως παραμετροποιημένο ώστε να μπορεί να χρησιμοποιηθεί σε πολλές περιπτώσεις διαφορετικών πολυμεσικών παρουσιάσεων. Η προσθήκη της βάσης δεδομένων ωφελεί στην επαναχρησιμοποίηση του συστήματος (reusability), πλεονέκτημα σημαντικό στον σύγχρονο κόσμο και στο διαδίκτυο.
- **Temporal links.** Μέσω των δυναμικά ορισμένων SMIL αρχείων, το σύστημα μπορεί και αναφέρεται σε ένα συγκεκριμένο χρονικό σημείο της ζωντανής.
- **Scriptability.** Η προσθήκη της γλώσσας PHP στην υλοποίηση του συστήματος παρέχει την εγγύηση ότι μελλοντικές πιο περίπλοκες καταστάσεις μπορούν να ξεπεραστούν με την προσθήκη κώδικα προγραμματισμού.
- **Web rendering.** Η απόδοση της αναπαραγωγής των πολυμέσων στο διαδίκτυο ήταν ικανοποιητική παρά τους περιορισμούς που ισχύουν όσον αφορά τη σχέση της γλώσσας SMIL και των browsers, όπως θα αναφέρουμε και στη συνέχεια.

- **Extensibility.** Το μοντέλο μας δεν περιορίζεται μόνο σε ζωντανές πολυμεσικές εφαρμογές αλλά, αντιθέτως, μπορεί να επεκταθεί και να καλύψει και άλλους τομείς λιγότερο αλληλεπιδραστικούς όπως για παράδειγμα είναι ένα διαδικτυακό quiz ή η ηλεκτρονική εκπαίδευση.

Επειδή η γλώσσα SMIL δεν υποστηρίζεται αυτούσια από τους browsers που υπάρχουν αυτή τη στιγμή στην αγορά χρησιμοποιήσαμε το plugin του Ambulant Player. Έτσι, η εφαρμογή δοκιμάστηκε στον Mozilla Firefox, σε Internet Explorer έκδοση 6 και σε Google Chrome. Παρουσιάστηκαν μερικά προβλήματα που οφείλονταν στο ότι plugin δεν ανταποκρινόταν σωστά κατά την εκτέλεση του SMIL αρχείου στον θεατή σε Mozilla Firefox και Internet Explorer ενώ στον Google Chrome ο χρήστης έπρεπε να κατεβάσει τα αρχεία SMIL τοπικά για να μπορεί να τα προβάλλει.

Εκτός από το πρόβλημα του περιορισμού στους browser, το σύστημά μας πρέπει να δοκιμαστεί περισσότερο και κάτω από πιο αντίξοες συνθήκες ιδιαίτερα όσον αφορά τις διαδικτυακές συνθήκες που θα επικρατούν. Πρέπει να γίνουν δοκιμές που θα αφορούν την πραγματική παρουσίαση μιας ζωντανής μετάδοσης μέσω live streaming και κάτω από συνθήκες χαμηλού bandwidth.

Τέλος, θεωρούμε ότι το επόμενο βήμα στην εξέλιξη του μοντέλου είναι η αναβάθμιση του προτύπου SMIL στην τελευταία έκδοση 3.0. Για να γίνει, βέβαια, κάτι τέτοιο πρέπει πρώτα να υποστηριχθεί από τις ανάλογες εμπορικές εφαρμογές, όπως να δημιουργηθούν τα αντίστοιχα browser plugins. Η τελευταία έκδοση προτύπου SMIL παρέχει νέες δυνατότητες που θα βοηθούσαν σημαντικά το σύστημά μας.

## 5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Το διαδίκτυο παρά τα πολλά χρόνια ωρίμανσής του δείχνει να μην αισθάνεται ακόμα άνετα όσον αφορά την διαχείριση των πολυμεσικών παρουσιάσεων. Έχουν διαπιστωθεί σοβαρά προβλήματα συγχρονισμού κατά το συνδυασμό πολυμεσικών αντικειμένων ειδικά σε ζωντανές διαδραστικές παρουσιάσεις. Ιδιαίτερο ενδιαφέρον παρουσιάζει και το πρόβλημα του late tuning στις ζωντανές μεταδόσεις.

Στην παρούσα αναφορά βασιστήκαμε σε μια δηλωτική προσέγγιση και χρησιμοποιήσαμε το πρότυπο SMIL ως τον κύριο άξονα για να πετύχουμε το συγχρονισμό και τη διαχείριση του χρόνου σε διαδικτυακές παρουσιάσεις. Μελετήσαμε το σενάριο ζωντανών πολυμεσικών παρουσιάσεων στο διαδίκτυο όπου είναι γνωστό ότι κάποια γεγονότα θα συμβούν αλλά δεν γνωρίζουμε ακριβώς πότε. Το μοντέλο που παρουσιάσαμε προσφέρει στον broadcaster την δυνατότητα να επιλέξει την κατηγορία του θεάματος και να ορίσει ποια γεγονότα μπορεί να εμφανιστούν σε αυτή. Κατόπιν, μπορεί να

δημιουργήσει δυναμικά replay, δηλαδή SMIL αρχεία, για καθένα από τα γεγονότα που θα συμβούν τη χρονική στιγμή που θα συμβούν μέσω της γλώσσας PHP.

Οι πληροφορίες αυτές καταγράφονται σε μια βάση δεδομένων με αποτέλεσμα να μπορούν να χρησιμοποιηθούν εύκολα στο μέλλον αλλά να γίνονται και αντικείμενο αναζήτησης στο διαδίκτυο. Παράλληλα, ο θεατής που συντονίστηκε ετεροχρονισμένα στη ζωντανή μετάδοση έχει την ευκαιρία να παρακολουθήσει τα γεγονότα που έχασε, λύνοντας το πρόβλημα του Late tuning.

Το σύστημα που προτείναμε δοκιμάστηκε σε διαφορετικά σενάρια με μεγάλη επιτυχία. Είναι παραμετροποιημένο με τέτοιο τρόπο που μπορεί να εφαρμοστεί σε όλες τις ζωντανές πολυμεσικές παρουσιάσεις στο διαδίκτυο. Θα ήταν εξαιρετικά ενδιαφέρον να δοκιμαστεί και σε συνθήκες live streaming αλλά και να προσαρμοσθεί ώστε να εφαρμοστεί και στην ψηφιακή τηλεόραση IDTV. Επίσης, μελλοντικά το μοντέλο θα μπορούσε να αναβαθμιστεί για να χρησιμοποιεί όλες τις νέες δυνατότητες που προσφέρει η γλώσσα SMIL 3.0. Πιστεύουμε ότι το βλέμμα πρέπει να στραφεί στο πρότυπο της SMIL έτσι ώστε το διαδίκτυο μαζί με τα δικά της πρότυπα, όπως την HTML5, XHTML και CSS3 να βρει μια ικανοποιητική πλατφόρμα λύσης για το συγχρονισμό και το πρόβλημα του late tuning στις ζωντανές πολυμεσικές παρουσιάσεις.

## ΑΝΑΦΟΡΕΣ

1. Dick C. A. Bulterman (2004) A linking and interaction evaluation test set for SMIL, Hypertext 04 Proceedings of the fifteenth ACM conference on Hypertext and hypermedia, pp. 46-47
  2. Ombretta Gaggi and Luca Danese (2011), A SMIL player for any web browser, In DMS 2011, pages 114–119
  3. W3C Synchronized Multimedia Working Group, GRiNS: A GRaphical Interface for Creating and Playing SMIL Documents
  4. Jack Jansen and Dick C. A. Bulterman (2009) Enabling Adaptive Time-based Web Applications with SMIL State
  5. Scherp A, Boll S (2004) Generic support for personalized mobile multimedia tourist applications.
- MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia (2004). doi: 10.1145/1027527.1027566
6. Boyer J (2007) XForms 1.0 (Third Edition). W3C. Available on: <http://www.w3.org/TR/xforms/>
  7. Hickson I (2007) XML Binding Language (XBL) 2.0. W3C. Available on: <http://www.w3.org/TR/xbl/>
  8. M. C. Maques Neto and C. A. S. Santos (2009), An approach based on events for treating the late tuning problem in interactive live TV shows, EiMM '09 Proceedings of the 1st ACM international workshop on Events in multimedia, pp. 41-48
  9. R.M. de Resende Costa, M.F. Moreno, R.F. Rodrigues, and L.F.G. Soares (2006), Live editing of hypermedia documents, DocEng '06: Proceedings of the 2006 ACM Symposium on Document Engineering. New York, NY, USA: ACM, 2006, pp.165-172
  10. M. C. Maques Neto and C. A. S. Santos (2008), An event-based model for interactive live TV shows, MM '08, Proceedings of the 16<sup>th</sup> ACM international conference on Multimedia, pp. 845-848
  11. F. Cavenaze, V. Quint and C. Roisin (2011), Timesheets.js: When SMIL Meets HTML5 and CSS3, DocEng '11: Proceedings of the 11th ACM Symposium on Document Engineering. New York, NY, USA: ACM, 2011, pp.43-52
  12. G. Blakowski and R. Steinmetz (1996), A media synchronization survey: reference model, specification, and case studies, IEEE Journal in Communications, Vol 14. No1, Jan. 1996 pp. 5-31
  13. Macromedia Flash Adobe Player, <http://www.macromedia.com/software/flash/about/>
  14. W3C, Synchronized Multimedia Integration Language, <http://www.w3.org/TR/smil/>
  15. Ambulant Open SMIL Player, <http://www.ambulantplayer.org/>

## APPENDIX

Αρχείο smil – content in advance που προβάλλει στατιστικές πληροφορίες πρωταθλήματος ποδοσφαιρικών αγώνων.

```
<?xml version="1.0"?>
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0//EN"
    "http://www.w3.org/2001/SMIL20/SMIL20.dtd">
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <meta name="title" content="Oscar 2013"/>
    <meta name="author" content="Vasiliki"/>
  </head>
  <layout>
    <root-layout id="Player-Window" backgroundColor="gray" width="610" height="430"/>
    <region id="audio"/>
    <region id="audio2"/>
    <region id="bkgd_image" width="610" height="400" />
    <region id="text" width="340" height="185" z-index="1" left="3"/>
    <region id="Video" left="23" width="550" top="13" height="228" z-index="2"/>
      <region id="Video1" left="13" width="520" top="60" height="228" />
    <region id="Menu" left="0" width="600" top="250" height="200" z-index="1"/>
  </layout>
  <transition id="slideover" type="slideWipe"/>
  <transition id="fade" type="fade"/>
  <transition id="push" type="pushWipe"/>
</head>
<body>
  <par id="Football replays">
    <seq>
      <par id="Intro" endsync="SkipIntro">
        <seq>
          
        </seq>
        <audio id="HmGeb" region="audio2" src="kalimba.mp3"/>
        <video id="HmGeb" region="Video1" src="xorigos.mp4"/>
        
      </par>
      <par id="MenuImages" dur="indefinite">
        <par>
          
            <area id="Ws" coords="7,5,175,110"/>
            <area id="Bs" coords="195,5,300,110"/>
            <area id="Fs" coords="400,5,570,110"/>
          </img>
          <excl id="Videos" dur="indefinite" fillDefault="freeze">
            <par id="Wz3" begin="Ws.activateEvent">
              <video id="W1s-0" region="Video" src="2.mp4"/>
              <audio id="Birthday-0" region="audio" src="kalimba.mp3"/>
            </par>
            <par id="Fz3" begin="Bs.activateEvent">
              <video id="Fz3-0" region="Video" src="3.mp4"/>
              <audio id="Birthday" region="audio" src="kalimba.mp3"/>
            </par>
            <par begin="Fs.activateEvent">
              <seq>
                
                <video id="BTz3" region="Video" fill="transition" src="4.mp4" transIn="fade"/>
                
                <video id="BPz3" region="Video" src="5.mp4" transIn="fade"/>
              </seq>
            </par>
          </excl>
        </par>
      </par>
    </seq>
  </par>
</body>
</smil>
```

```

        </par>
    </excl>
</par>
</par>
</seq>

</par>
</body>
</smil>

```

## My\_functions.php

```

<?php

//συνάρτηση για συνδεση με τη βαση δεδομένων
function connect_db()
{
    $con = mysql_connect("localhost","db_smil","1234");
    if (!$con)
    {
        die('Could not connect: ' . mysql_error());
    }

    mysql_select_db("db_smil") or die ("Couldn't select the database.");

    mysql_query("SET CHARACTER SET 'utf8'");
    mysql_query("SET NAMES 'utf8'");
    return $con;
}

?>

```

## Header.php

```

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
        <meta http-equiv="Content-Language" content="el">

        <link rel='stylesheet' href='./css/all.css' type='text/css' />
        <style type='text/css'>
        </style>
    </head>
    <body>

```

## Broadcaster.php

```

<?php

require("myfunctions.php");
require("header.php");
$con=connect_db();
mysql_query("SET CHARACTER SET 'utf8'");
mysql_query("SET NAMES 'utf8'");

$date = date('Y-m-d');
$show_date = date('d-m-Y');
echo $show_date;

```

```

//αν ο broadcaster έχει επιλέξει να προσθεσει μια νεα κατηγορια παρουσιασης-θεαματος
if (isset($_GET['do_add_show_type']))
{if ( $_GET['do_add_show_type'] == 1 )
{
    $description_newshowtype = $_GET['description_newshowtype'];
    $result1 = @mysql_query("SELECT * FROM show_type WHERE type_description='$description_newshowtype'");
    $row = mysql_num_rows($result1);

    //ελεγχος αν εχουν συμπληρωθει ολα τα πεδια
    if( ($row > 0) || (trim($description_newshowtype) == "") )
    {
        echo "
        <div id='header'>";
        echo "<span style='font-size:13pt;'>Υπάρχει κατηγορία θεάματος με το ίδιο όνομα ή τα στοιχεία δεν είναι
        συμπληρωμένα όλα σωστά.</span><br>";
        echo "-----<br>";
        echo "</div>";
    }
    else
    {
        //εισαγωγή νεας κατηγοριας θεαματος στην βαση δεδομενων
        $insert = mysql_query("INSERT INTO show_type (type_description)
        VALUES ('{$description_newshowtype}')");

        if(!$insert)
            echo "<font color='red'><h3>Παρουσιάστηκε πρόβλημα.</h3>Δεν είναι δυνατή η εισαγωγή στοιχείων
            στην βάση δεδομένων.</font><p>";
        else
        {
            echo "
            <div id='header'>";
            echo "<span style='font-size:13pt;align:center;'>Επιτυχής καταχώρηση κατηγορίας θεάματος.
            </span><br>";
            echo "-----<br>";
            echo "</div>";
        }
        @mysql_free_result($insert);    }
    }
}

//αν ο broadcaster έχει επιλέξει να προσθεσει ενα συμβαν(event) σε μια κατηγορια θεαματος
if (isset($_GET['do_add_show_event']))
{
    if ( $_GET['do_add_show_event'] == 1 )
    {
        $type_id_event = $_GET['type_id_event'];
        $description_newshowevent = $_GET['description_newshowevent'];

        $result2 = mysql_query("SELECT * FROM events WHERE type_id=$type_id_event AND
        events_name='$description_newshowevent'");

        //ελεγχος αν εχουν συμπληρωθει ολα τα πεδια
        if(($type_id_event == "") or (trim($description_newshowevent) == "") or (mysql_num_rows($result2)>0))
        {
            echo"<div id='header'>";
            echo "<span style='font-size:13pt;'>Υπάρχει event με το ίδιο όνομα ή τα στοιχεία δεν είναι
            συμπληρωμένα όλα σωστά. </span><br>";
            echo "-----<br>";
            echo "</div>";
        }
        else
        {
            //εισαγωγή νεου event σε μια κατηγορια θεαματος στην βαση δεδομενων
            $insert = mysql_query("INSERT INTO events (events_name, type_id)
            VALUES ('{$description_newshowevent}','{$type_id_event}')");

            if(!$insert)

```

```

        echo "<font color='red'><h3>Παρουσιάστηκε πρόβλημα.</h3>Δεν είναι δυνατή η εισαγωγή  
στοιχείων στην βάση δεδομένων.</font><p>";
    else
    {
        echo "<div id='header'>";
        echo "<span style='font-size:13pt;'>Επιτυχής καταχώρηση event.</span><br>";
        echo "-----<br>";
        echo "</div>";
    }
    @mysql_free_result($insert);
}
}
}
//προβολή φορμας του broadcaster για επιλογή του ειδους παρουσιασης-θεαματος live streaming
//και εισαγωγής χρησιμων πληροφοριων για αυτο(οπως το Link της ζωντανής μετάδοσης)
echo"
<div id='header'>
<form method='GET' action='set_parametres.php' >
<input type=hidden name='do_add' value= 1 >;
$result = mysql_query("SELECT * FROM show_type ORDER BY type_description ASC;");
echo"<font size=4>Κατηγορία Θεάματος:</font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~";
<select name='type_id'>
<option value="" >-Επιλέξτε-</option>";

while ($row = mysql_fetch_array($result) )
{
    echo"
        <option value={$row['type_id']} >{$row['type_description']}";
}
@mysql_free_result($result);

echo"
</select><br>
<span style=' margin-left:14px;margin-right:-98px;font-size:12pt;'>Περιγραφή-Σχόλια:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</span>
<input type='text' name='description' value="" size='30' style='margin-left:120px; margin-top:10px;'><br/>
<span style='margin-left:216px; font-size:12pt;'>Link live streaming: <span>
<input type='text' name='link_name' value="" size='45' style='margin-left:20px; margin-top:10px;'>
<input type='submit' value='Συνέχεια' style='margin-left:50px;'>
</form>
</div>";

//προβολή φορμας για εισαγωγή νεας κατηγοριας παρουσιασης-θεαματος απο τον broadcaster
echo"
<div id='manage_show_type'>
<br><form id='add'>
<a href='broadcaster.php?add1=1' onclick='add(2)'><img src='plus.png'/></a>
    Προσθήκη νέας Κατηγορίας Θεάματος <div>";

if (isset($_GET['add1']))
{
    $add1=$_GET['add1'];

    echo"<div id='show_type'>
    <form method='GET' action='broadcaster.php' >
    <input type=hidden name='do_add_show_type' value= 1>;

    echo"<br>
    <span>Περιγραφή :</span>
    <input type='text' name='description_newshowtype' value="" size='45' style='margin-left:20px;
        margin-top:10px;'>
    <input type='submit' value='Προσθήκη' style='margin-left:50px;'>
    </form>
    </form>
    </div>";
}

//προβολή φορμας για εισαγωγή νευ event σε μια κατηγορια παρουσιασης-θεαματος

```



```
echo"
<br>
<div id='manage_show_type'>
<br><form id='add'>
<a href='broadcaster.php?add2=1' onclick='add(2)'><img src='plus.png'/></a>
    Προσθήκη νέου event σε κατηγορία θεάματος";

echo"
</div>;
if (isset($_GET['add2']))
{
    $add2=$_GET['add2'];
    echo"
    <div id='show_type'><br>
    <form method='GET' action='broadcaster.php' >
    <input type='hidden' name='do_add_show_event' value= 1>;
    echo"<br>";
    $result = mysql_query("SELECT * FROM show_type ORDER BY type_description ASC;");
    echo"Κατηγορία Θεάματος:</font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~";
    <select name='type_id_event'>
    <option value="" >-Επιλέξετε-</option>;

    while ($row = mysql_fetch_array($result) )
    {
        echo"
            <option value={$row['type_id']} >{$row['type_description']}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~";
        }
    @mysql_free_result($result);

    echo"</select><br>
    <span>Περιγραφή :</span>
    <input type='text' name='description_newshowevent' value="" size='45' style='margin-left:20px; margin-top:10px;'>
    <input type='submit' value='Προσθήκη' style='margin-left:50px;'>
    </form>
    </form>
    </div>;
}
echo"
</body>
</html>

";
?>
```

## Set\_parametres.php

```
<script language="JavaScript" type="text/javascript">

function form_flag() {
var returnval
var number = /^[0-9 ]*$/;

var returnval= document.getElementById("file").value;
return returnval;

if((document.profile.videoname.value=="") )
{
alert("Παρακαλούμε συμπληρώστε το πεδίο Όνομα Video ");

returnval=false
return returnval;

}

}
```

```

if((document.profile.description.value=="") )
{
    alert("Παρακαλούμε συμπληρώστε το πεδίο Περιγραφή ");

    returnval=false
    return returnval;
}

if(!document.profile.firsthour.value.match(number) ||!document.profile.firstminutes.value.match(number) || !
document.profile.firstseconds.value.match(number))
{
    alert("Παρακαλούμε συμπληρώστε αριθμό για τα πεδία χρόνου έναρξης του video-replay");

    returnval=false
    return returnval;
}

if(!document.profile.lasthour.value.match(number) ||!document.profile.lastminutes.value.match(number) || !
document.profile.lastseconds.value.match(number))
{
    alert("Παρακαλούμε συμπληρώστε αριθμό για τα πεδία χρόνου λήξης του video-replay");

    returnval=false
    return returnval;
}

if ( (document.profile.firsthour.value=="") && (document.profile.firstminutes.value=="") &&
(document.profile.firstseconds.value=="") ) {

    alert("Παρακαλούμε συμπληρώστε τον αριθμό έναρξης του video-replay");

    returnval=false
    return returnval;
}
if ( (document.profile.lasthour.value=="") && (document.profile.lastminutes.value=="") &&
(document.profile.lastseconds.value=="") ) {

    alert("Παρακαλούμε συμπληρώστε τον αριθμό λήξης του video-replay");

    returnval=false
    return returnval;
}
}

var my_date=new Date();

</script>
<?php

    session_start();
    require("myfunctions.php");
    require("header.php");
    $con=connect_db();
    mysql_query("SET CHARACTER SET 'utf8'");
    mysql_query("SET NAMES 'utf8'");

    $date = date('Y-m-d');
    $show_date = date('d-m-Y');
    echo $show_date;

    //αν έχει πατηθεί το κουμπι για την δημιουργια ενος video-relay
    if(isset($_GET['do_add']))

```

```
{
if ($_GET['do_add']==1)
{
$type_id = $_GET['type_id'];
$description = $_GET['description'];
$link_name= $_GET['link_name'];
//βρισκουμε , αν υπαρχουν, shows στην βαση δεδομενων με τα ιδια κριτηρια επιλογης του broadcaster
$result = mysql_query("SELECT * FROM shows WHERE type_id=$type_id AND show_name='$description' AND date='$date' ;");
//αν δεν εχει συμπληρωθει σωστα η φορμα ή αν εχει βρεθει εγγραφη με τα ιδια στοιχεια, παρουσιαζεται μηνυμα σφαλματος
if( ($type_id == "") or (trim($description) == "") or (trim($link_name) == "") or mysql_num_rows($result)>0 )
{
echo "<div id='header'>";
echo "<span style='font-size:13pt;'>Υπάρχει εγγραφή με το ίδιο όνομα ή τα στοιχεία δεν είναι συμπληρωμένα όλα σωστά.</span><br>";
echo "-----<br>";
echo"</div>";
}
else
{
//εισαγωγη show στην βαση δεδομενων
$insert = mysql_query("INSERT INTO shows (date,show_name,show_comments,type_id,link_name) VALUES (NOW(),'{ $description}','',{ $type_id},{ $link_name}')");

if(!$insert)
echo "<font color='red'><h3>Παρουσιάστηκε πρόβλημα.</h3>  
Δεν είναι δυνατή η εισαγωγή στοιχείων στην βάση δεδομένων.</font><p>";
else
{
//echo "<h3>Επιτυχής καταχώρηση αρχείου</h3><br>";
}
@mysql_free_result($insert);

//προβολη της σελιδας με την live streaming παρουσιαση, οπου δινεται η δυνατότητα στον broadcaster να δημιουργήσει video-replay
echo"<div id='header'>";
$result2 = mysql_query("SELECT * FROM show_type WHERE type_id=$type_id ;");
echo"<font size='4'>Κατηγορία Θεάματος:</font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~";
$row2 = mysql_fetch_array($result2);
echo "<font size='3'><b>{$row2['type_description']}</b></font>";
@mysql_free_result($result2);

echo "<br/>";
$result3 = mysql_query("SELECT * FROM shows ORDER BY show_id DESC LIMIT 1;");
$row3 = mysql_fetch_array($result3);
echo "<font size='3'><b>{$row3['show_name']}</b></font>";
@mysql_free_result($result3);

echo"
</div>
<div id='main'>
<table width='100%' style='margin-top:-115px;'>
<tr>
<td >
<video width=\"800\" height=\"310\" controls >
<source src= '{ $row[\'link_name\']}' >
</video>
</td>
<td >
<form method='POST'
action='write_file.php?
show_id={$row3['show_id']}&link_name={$row3['link_name']}&type_id=$type_id'
name='profile' onSubmit='return form_flag()' target='_blank' enctype='multipart/form-data'>;

$result1 = mysql_query("SELECT * FROM events WHERE events.type_id=$type_id ORDER BY events name ASC.");
```



```

<form method='GET' action='set_parametres1.php' >
<input type=hidden name='do_add' value= 1 >";
$result = mysql_query("SELECT * FROM show_type ORDER BY type_description ASC;");
echo"<font size=4>Κατηγορία Θεάματος:</font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<select name='type_id'>
<option value=" >-Επιλέξτε-</option>";

while ($row = mysql_fetch_array($result) )
{
    echo"
        <option value={$row['type_id']} >{$row['type_description']}";
    }
}
@mysql_free_result($result);

echo"
</select><br>
<span style=' margin-left:14px;font-size:12pt;'>Περιγραφή-Σχόλια:</span>
    <input type='text' name='description' value=" size='30' style='margin-left:20px; margin-top:10px;'>
<br><span style='margin-left:216px; font-size:12pt;'>Live streaming link:</span>
    <input type='text' name='link_name' value=" size='45' style='margin-left:20px; margin-top:10px;'>
<input type='submit' value='Συνέχεια' style='margin-left:50px;'>
</form>
</div>";

echo"
</body>
</html>
";
?>

```

## Write\_file.php

```

<?php

session_start();
require("myfunctions.php");
require("header.php");
$con=connect_db();
mysql_query("SET CHARACTER SET 'utf8'");
mysql_query("SET NAMES 'utf8'");

$link_name=$_GET['link_name'];
$type_id=$_GET['type_id'];
$show_id=$_GET['show_id'];
$events_id=$_POST['events_id'];
$description=$_POST['description'];
$videoname=$_POST['videoname'];
$firsthour=$_POST['firsthour'];
$firstminutes=$_POST['firstminutes'];
$firstseconds=$_POST['firstseconds'];
$lasthour=$_POST['lasthour'];
$lastminutes=$_POST['lastminutes'];
$lastseconds=$_POST['lastseconds'];
//το μονοπατι για τα video extra περιεχομενου, οπως video με την κερκιδα των ομαδων
$target_path_contentinadvance='./content_in_advance/';

//αν ο broadcaster δεν εχει συμπληρωσει καποια απο τα πεδια των χρονων εναρξης-ληξης των video, τοτε συμπληρωνεται
αυτοματα ο αριθμος 00
if ($firsthour=="")
{
    $firsthour=00;
}
if ($firstminutes=="")

```

```

{
    $firstminutes=00;
}
if ($firstseconds==" )
{
    $firstseconds=00;
}
if ($lasthour==" )
{
    $lasthour=00;
}
if ($lastminutes==" )
{
    $lastminutes=00;
}
if ($lastseconds==" )
{
    $lastseconds=00;
}

$result = mysql_query("SELECT show_name FROM shows WHERE show_id=$show_id;");
$row = mysql_fetch_array($result);
@mysql_free_result($result);
$show_name= $row['show_name'];
$description = mb_convert_encoding($description, "ISO-8859-7", "UTF-8");
//ορισμος του μονοπατιου που αποθηκευονται τα video-replay smil καθώς και τα txt αρχεία με πληροφορίες του video
$target_path1 = './uploads/';
$files_extension = 'smil';
$files_extension_information = 'txt';

$videoname = mb_convert_encoding($videoname, "ISO-8859-7", "UTF-8");
$description = mb_convert_encoding($description, "ISO-8859-7", "UTF-8");
$target_path = $target_path1 . $description . '.' . $files_extension;

$target_path_information = $target_path1 . $description . '.' . $files_extension_information;

//δημιουργία txt αρχείων με πληροφορίες του video smil
$file_information = fopen("$target_path_information", "w");
echo fwrite($file_information, " $show_name
'$description'");
fclose($file_information);

//αν ο broadcaster δεν έχει επιλέξει αν προσθεσει στο video-replay αρχείο extra περιεχομενου
if ($_FILES['file']['error'] > 0)
{
    //δημιουργια smil αρχείου σύμφωνα με τις παραμετρους που δίνει ο broadcaster
    $file = fopen("$target_path", "w");
    echo fwrite($file, "<smil xmlns='http://www.w3.org/2001/SMIL20/'>
<head>
<layout>
<root-layout width='520' height='480' />
<region id='a' top='50' height='500' width='550' z-index='2' />
<region id='b' top='5' regalign='left' background-color='white' />
</layout>
</head>
<body>
<par>
<video src='$link_name' region='a' clipBegin='npt=$firsthour:$firstminutes:$firstseconds'
clipEnd='npt=$lasthour:$lastminutes:$lastseconds'>
</video>
<text src='$description.txt' region='b' />
</par>
</body>
</smil>
");
fclose($file);
}

```

```

//αν ο broadcaster έχει επιλέξει αν προσθέσει στο video-replay αρχείο extra περιεχομένου
else
{
    $filename= $target_path_contentinadvance.$_FILES['file']['name'];
    //dimourgia smil video simfona me tis parametrous pou dinei o broadcaster
    $file = fopen("$target_path","w");
    echo fwrite($file,"<smil xmlns='http://www.w3.org/2001/SMIL20/'>
    <head>
    <layout>
    <root-layout width='520' height='480' />
        <region id='a' top='50' height='500' width='550' z-index='2' />
        <region id='b' top='5' regalain='left' background-color='white' />
    </layout>
    </head>
        <body>
        <par>
            <video src='$link_name' region='a' clipBegin='npt=$firsthour:$firstminutes:$firstseconds'
clipEnd='npt=$lasthour:$lastminutes:$lastseconds'>
            </video>
            <text src='$description.txt' region='b' />
        </par>
        <seq>
            <video src='$filename' region='a'>
            </video>
        </seq>
        </body>
    </smil>
    ");
}

//αν υπάρχει ήδη τετοιο video στην βαση δεδομενων
if (file_exists("$target_path"))
{
    if ( 0 == filesize( "$target_path" ) )
    {
        //αν το smil video ειναι αδειο
        exit("file is empty");
    }
    else
    {
        $time_first =$firsthour.'.'. $firstminutes.'.'. $firstseconds;

        //εισαγωγή στην βαση δεδομενων μονοπατιου και παραμετρων του smil video
        $insert = mysql_query("INSERT INTO attributes
        (attr_description,path_smil,time_first,show_id,events_id,name_video,old)
        VALUES ('{$description}','{$target_path}','{$time_first}','{$show_id}','{$events_id}',
        '{$description}','0');");

        if(!$insert)
            echo "<font color='red'><h3>Παρουσιάστηκε πρόβλημα.</h3>
            Δεν είναι δυνατή η εισαγωγή στοιχείων στην βάση δεδομένων.</font><p>";
        else
        {
            echo "<h3>Επιτυχής καταχώρηση αρχείου</h3><br>";
        }
        @mysql_free_result($insert);
    }
}

echo"
</body>
</html>";
?>

```

## Provoli\_smil\_files.php

```
<?php

session_start();
require("myfunctions.php");
require("header.php");
$con=connect_db();
mysql_query("SET CHARACTER SET 'utf8'");
mysql_query("SET NAMES 'utf8'");

//αν έχει πατηθεί το κουμπι διαγραφη
if(isset($_GET['do_delete']))
{
    if ($_GET['do_delete']==1)
    {
        // id attribute-video προς διαγραφη
        $attributes_id = $_GET['attributes_id'];

        $result3 = mysql_query("SELECT *FROM attributes
                                WHERE attributes_id={ $attributes_id } ;");

        if(mysql_num_rows($result3)>0)
        {
            while($row3 = mysql_fetch_array($result3))
            {
                //διαγραφη αρχιων(smil και txt)απο τον φακελο
                //uploads, που αναφέρονται στο συγκεκριμενο attribute
                @unlink('./uploads/' . $row3['name_video'].'.smil');
                @unlink('./uploads/' . $row3['name_video'].'.txt');
            }
        }
        @mysql_free_result($result3);

        //διαγραφη αρχιου απο την βαση δεδομενων
        $result = mysql_query("DELETE FROM attributes WHERE attributes_id = { $attributes_id };");

        if(!$result)
            echo "<font color='red'><h3>Παρουσιάστηκε πρόβλημα.</h3>
                    Δεν είναι δυνατή η ολοκλήρωση του ερωτήματος.</font><p>";
        else
            echo ' <h4>Επιτυχής Διαγραφή Στοιχείων</h4>';

        @mysql_free_result($result);
    }
    echo "-----<br>";
}

//αν έχει πατηθεί η επιλογή "Προσθηκη στον θεατη", οπου το συγκεκριμενο attribute στέλνεται στην λιστα του θεατη
if(isset($_GET['do_add_viewer']))
{
    if ($_GET['do_add_viewer']==1)
    {
        // id attribute-video προς μεταβαση στην λιστα του θεατη
        $attributes_id = $_GET['attributes_id'];
        $insert = mysql_query("INSERT INTO attributes
(attr_description,path_smil,time_first,show_id,events_id,name_video,old) SELECT
attr_description,path_smil,time_first,show_id,events_id,name_video,l FROM attributes WHERE
attributes_id={ $attributes_id } ;");
    }
}
```



```

        if(!$insert)
            echo "<font color='red'><h3>Παρουσιάστηκε πρόβλημα.</h3>
            Δεν είναι δυνατή η εισαγωγή στοιχείων στην βάση δεδομένων.</font><p>";
        else
        {
            //echo "<h3>Επιτυχής καταχώρηση αρχείου</h3><br>";
        }
        @mysql_free_result($insert);
    }
    echo "-----<br>";
}

//αν έχει πατηθεί η αναζήτηση smil video σύμφωνα με τα κριτηρια επιλογής του broadcaster
if(isset($_GET['do_search']))
{
    if ( $_GET['do_search'] == 1 )
    {

        $year=$_GET['year'];
        $month=$_GET['month'];
        $day=$_GET['day'];
        //imerominia
        $date=$year.'-'. $month.'-'. $day;

        $search_text=$_GET['search_text'];

        //αν δεν έχει συμπληρωθεί το πεδίο της αναζήτησης
        if ($search_text==" ")
            $where_text=" ";

        //αν δεν έχει συμπληρωθεί το πεδίο της αναζήτησης
        if ($search_text!=" ")
            $where_text=" and (show_name LIKE '%$search_text%' or attr_description
            LIKE '%$search_text%')";

        //αν δεν έχει δώσει η ημερομηνία ως κριτήριο αναζήτησης
        if ($month==0 && $day==0 && $year==0)
            $where_date=" ";

        //αν έχει δώσει η ημερομηνία ως κριτήριο αναζήτησης
        if ($month!=0 && $day!=0 && $year!=0)
            $where_date=" and shows.date='$date' ";

        $result = mysql_query("SELECT * FROM shows,attributes
        WHERE shows.show_id=attributes.show_id $where_date $where_text ");
        while ($row = mysql_fetch_array($result))
        {
            $result1 = mysql_query("SELECT events_name FROM events WHERE
            events.events_id='{ $row['events_id'] }';");
            $row1 = mysql_fetch_array($result1);
            @mysql_free_result($result1);
            echo"<a href='{ $row['path_smil'] }' target='iframe_a'>
            { $row['show_name'] } - { $row1['events_name'] } - { $row['attr_description'] }</a>

            <form method='get' action='provoli_smil_files.php'>
                <input type='hidden' name='attributes_id' value='{ $row['attributes_id'] }'>
                <input type='hidden' name='do_delete' value=1>
                <input type='submit' value='Διαγραφή'>
            </form><form method='get' action='provoli_smil_files.php'>
                <input type='hidden' name='attributes_id' value='{ $row['attributes_id'] }'>
                <input type='hidden' name='do_add_viewer' value=1>
                <input type='submit' value='Προσθήκη στον θεατή'>
            </form>
            <br>";
        }
    }
}

```

```

    }
    if(mysql_num_rows($result)==0)
    {
        echo "Δεν υπάρχουν video με αυτά τα κριτήρια<br>";
    }
    @mysql_free_result($result);
}
echo "-----<br>";
}

//προβολη φορμας αναζητησης
echo "<b>Αναζήτηση video-replay</b> <br>";

echo"
<form method='get' action='provoli_smil_files.php' target='_self' enctype='multipart/form-data'>

<input type='text' name='search_text' value=' ' size='20' style='margin-bottom:5px; margin-top:5px;'><br>
<select name='month'>
    <option value='0'> --
    <option value='01'>Ιανουάριος
    <option value='02'>Φεβρουάριος
    <option value='03'>Μάρτιος
    <option value='04'>Απρίλιος
    <option value='05'>Μάιος
    <option value='06'>Ιούνιος
    <option value='07'>Ιούλιος
    <option value='08'>Αύγουστος
    <option value='09'>Σεπτέμβριος
    <option value='10'>Οκτώβριος
    <option value='11'>Νοέμβριος
    <option value='12'>Δεκέμβριος
</select>
<select name='day'>
    <option value='0'> --
    <option value='01'>1
    <option value='02'>2
    <option value='03'>3
    <option value='04'>4
    <option value='05'>5
    <option value='06'>6
    <option value='07'>7
    <option value='08'>8
    <option value='09'>9
    <option value='10'>10
    <option value='11'>11
    <option value='12'>12
    <option value='13'>13
    <option value='14'>14
    <option value='15'>15
    <option value='16'>16
    <option value='17'>17
    <option value='18'>18
    <option value='19'>19
    <option value='20'>20
    <option value='21'>21
    <option value='22'>22
    <option value='23'>23
    <option value='24'>24
    <option value='25'>25
    <option value='26'>26
    <option value='27'>27
    <option value='28'>28
    <option value='29'>29
    <option value='30'>30
    <option value='31'>31
</select>

```





```

echo"
<html>
  <head>
    <meta http-equiv='Content-Type' content='text/html; charset=utf-8'/>
    <meta http-equiv='Content-Language' content='el'>
    <meta http-equiv='refresh' content='40' >

    <link rel='stylesheet' href='./css/all.css' type='text/css' />
    <style type='text/css'>
    </style>
  </head>
  <body>";
  $date = date('Y-m-d');

  $show_id = $_GET['show_id'];

  //προβολη των video που εχουν ηδη δημιουργηθει απο τον broadcaster
  echo"
  <b><span style='font-size:12pt;'>Τα Highlights του θεάματος είναι τα εξής:</span></b><p>";

  $result = mysql_query("SELECT * FROM shows,attributes WHERE attributes.show_id=$show_id AND
    attributes.show_id=shows.show_id AND shows.date='$date' AND attributes.old='0';");
  $row_number=mysql_num_rows($result);
  //ελεγχος αν υπαρχουν video που εχουν δημιουργηθει από τον broadcaster για το συγκεκριμενο θεαμα
  if (mysql_num_rows($result) == 0)
  {
    echo "Δεν υπάρχουν διαθέσιμα highlights μέχρι την συγκεκριμένη χρονική στιγμή.";
  }
  else
  {
    $count=0;
    while ($row = mysql_fetch_array($result) )
    {
      $count++;
      $result1 = mysql_query("SELECT * FROM events WHERE events_id={ $row['events_id'] };");
      $row1 = mysql_fetch_array($result1);
      @mysql_free_result($result1);
      echo "<br><a href='{ $row['path_smil'] }' target='iframe_b'>
        { $row['show_name'] } - { $row1['events_name'] } - { $row['attr_description'] }</a>";

      if($count==$row_number)
      {
        echo " - new";
      }
    }
    @mysql_free_result($result);
  }

  echo"
  </body>
</html>";
?>

```

## Provoli\_old.php

```

<?php

session_start();
require("myfunctions.php");
require("header.php");
$con=connect_db();
mysql_query("SET CHARACTER SET 'utf8'");
mysql_query("SET NAMES 'utf8'");

```

```

//id του show
$show_id=$_GET['show_id'];

//link στο smil αρχείο με τα στατιστικά του πρωταθλήματος
echo"<a href='./uploads/statistics.smil' target='iframe_b'>Στατιστικά Πρωταθλήματος</a><br><br>";

echo"
    <b><span style='font-size:12pt;'>Δείτε επίσης:</span></b><br><br>";

//προβολή των video-replay παλιών παρουσιάσεων που ο broadcaster επέλεξε να προσθεσει στην λιστα του θεατή
$result = mysql_query("SELECT * FROM shows,attributes WHERE shows.show_id=attributes.show_id AND
    attributes.show_id=$show_id AND attributes.old=1 ;");
while ($row = mysql_fetch_array($result))
{
    $result1= mysql_query("SELECT show_name FROM shows WHERE show_id='{ $row['show_id']}' ;");
    $row1 = mysql_fetch_array($result1);
    @mysql_free_result($result1);
    echo"<a href='{ $row['path_smil']}' target='iframe_b'>{ $row1['show_name']} - { $row['attr_description']}</a> <br>";

}
//αν δεν υπάρχουν video προς εμφάνιση
if(mysql_num_rows($result)==0)
{
    echo "Δεν υπάρχουν video με αυτά τα κριτήρια<br>";
}
@mysql_free_result($result);

echo"
</body>
</html>";
?>

```