# Component-based Hypervideo Model:  High Level Specification of Hypervideos

**Madjid SADALLAH** – *CERIST, Algeria*

**Olivier AUBERT** – *LIRIS, France*

**Yannick PRIÉ** – *LIRIS, France*

# Context & Background

# Hypervideo

- Video-based information nowadays invasive → Increasing importance of AV-based documents

- How to better communicate AV-based data (knowledge)?

- Hypervideo: <span style="color:red">Video-centric</span> <span style="color:blue">Hypermedia</span> document

  - a main audiovisual content

  - metadata that augment the AV in a time synchronized way

  - new presentation and interaction modalities

# Main Hypervideo Specifities

- No video intrinsic structure + Need to break video linearity

  - → **Annotations** mandatory to describe, augment and address AV content/structure

- Rhetorical and aesthetic challenges

  - Variety of visualization modalities

- More pronounced hypermedia concerns:

  - Disorientation

  - Cognitive overload (e.g time pressure on the reader)

# Background

- Hypervideo ideas since Ted Nelson (1960s). Initial concepts: HyperCafe (1990s)

- Recently: Hyper-Hitchcock, Advene, HVet, VideoClix, Popcorn.js

- Languages: SMIL, NCL

- Models:

  - General hypermedia (DHM, AHM, NCM)

  - Restrictive and specific (Detail-on-Demand)

# Background

- Conventional hypermedia models:
    - Not hypervideo specific with concepts that are very general
    - Not much attention for hypervideo support
        - → do not describe precisely/clearly hypervideo properties
- Many existing HV implementations
    - Use informal/no HV models
    - Have technically driven representations
- Need for a dedicated and implementable model for hypervideos

# CHM:
# Component-based
# Hypervideo Model

# Principles

- CHM Purpose:

  - conceptually analyze existing hypervideos

  - help the design of new ones

- CHM:  presentation of synchronized AV metadata through nested low- and high-level components

- A component = building block that represents a formal information and composition unit

- A hypervideo = set of low and high level components

# General Overview



- Annotation is at the core

  - Structures and readers

- At least one *TimedMedia* element that addresses a temporalized stream

- TimedMedia played through a player component defines a *TimeLine Reference* (TLR)

- The TLR synchronizes the rendering of the related components (*TLComponents*)

- Multiple players and TLRs may co-exist → different hypervideo sub-documents
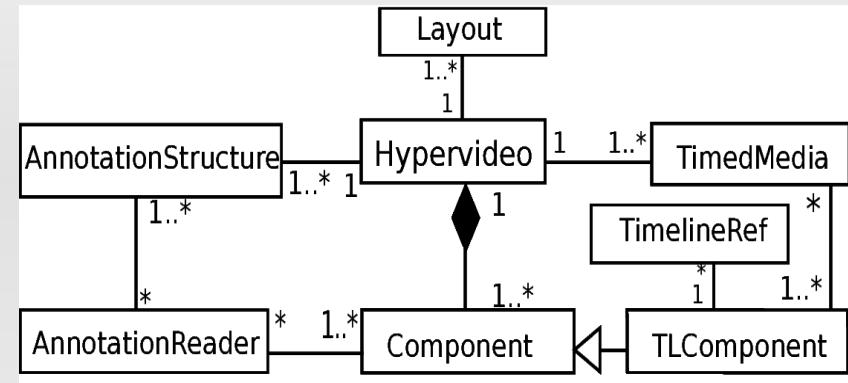
# CHM Annotation Model

- Complies with Advene/Cinelab Model

- Annotation = Any information (data/resource) associated to a spatio-temporal video fragment

- Annotation structure:

  - Video reference

  - Start/end timecodes

  - Type

  - Content

# CHM Plain Components
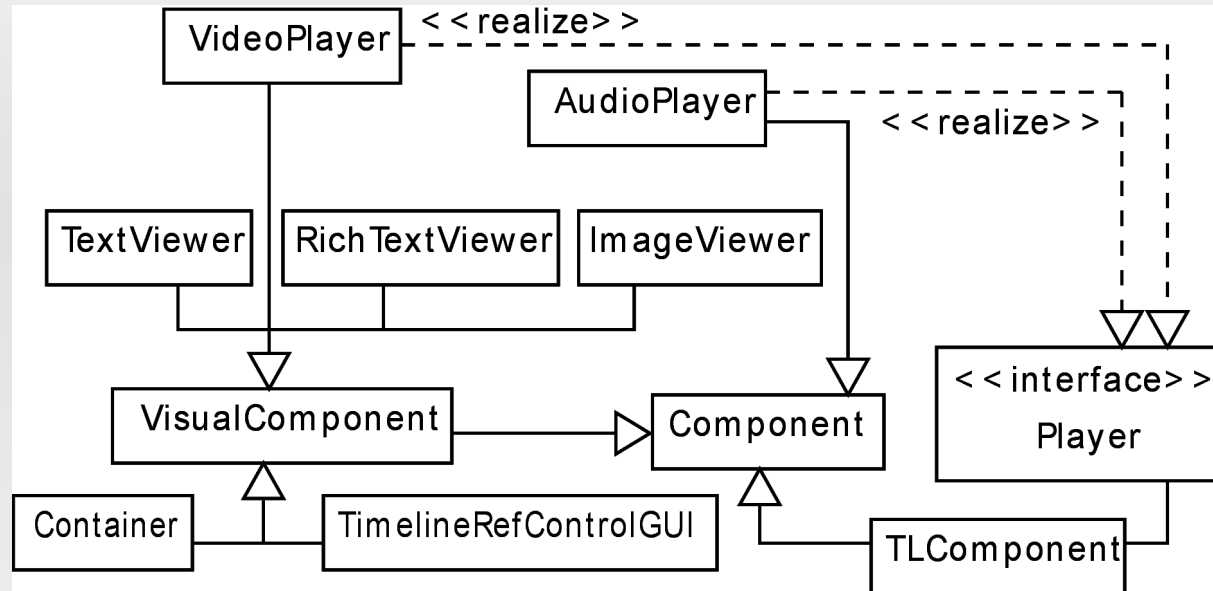


- Basic hypervideo data components

  – With interactive interfaces for rendering temporalized data

- Data provided as annotations, accessed through *AnnotationReader* components

# CHM High Level Components
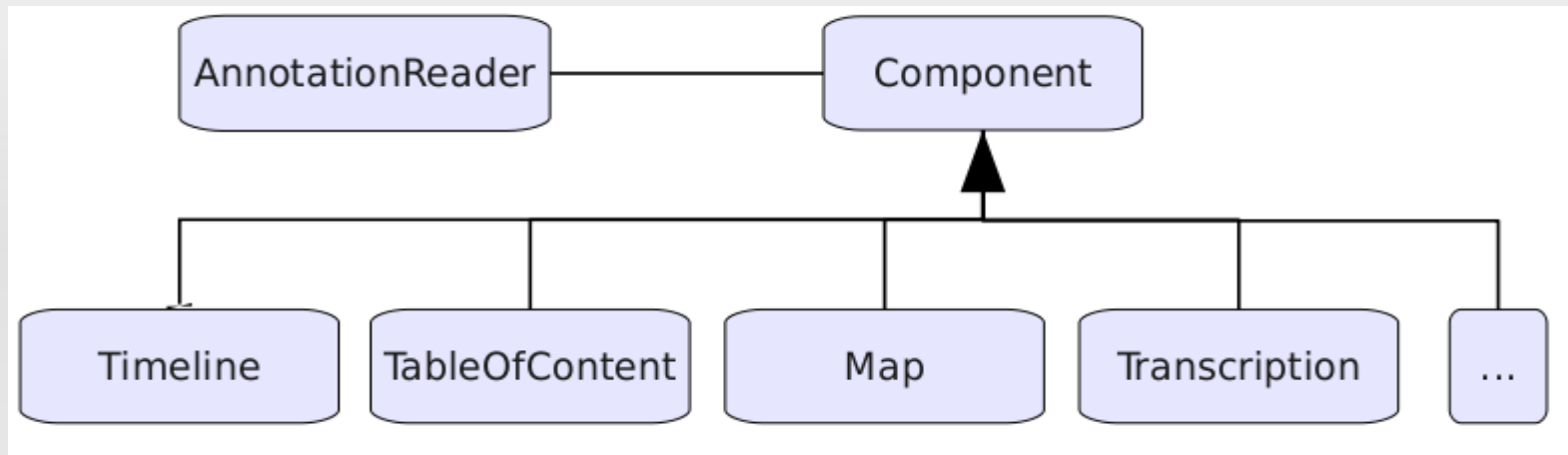


- Identified in many existing HV examples

- Built upon the plain components

- CHM supports these very common and useful set

- Extensible: from the plain ones or a combination plain/high

# CHM Spatial Model
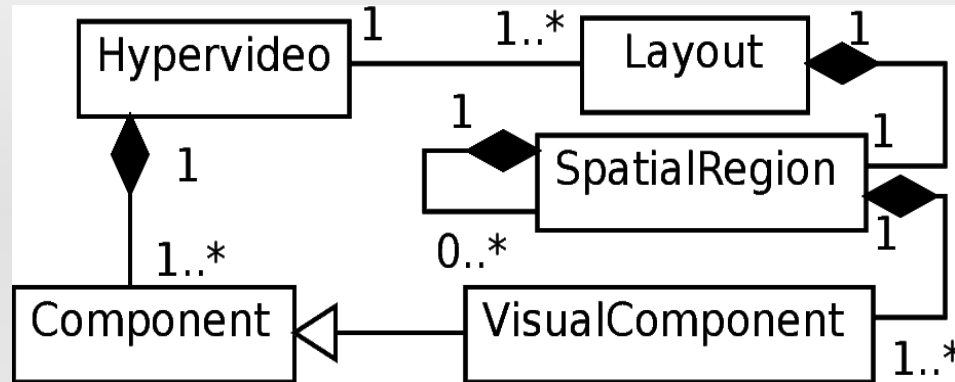


- Intends to accommodate the implementation platform specificities

- *VisualComponents* placed within *SpatialRegions*

- SpatialRegions global placement defined in the root *Layout* element

# CHM Temporal Model

- Timeline-based = explicit time scale

- TLR = virtual time reference

  – Linked to a player (that renders a TimedMedia component) or to the global document

  – Access and control: "state" , "position" and "duration" attributes

- TLR allows to synchronize components

- Update of the TLR position or state → Update of all the related TLComponents

# CHM Links and Events

- CHM hypervideo links

  - Unidirectional with no separate link components (SMIL/HTML, unlike AHM)

  - space and time (hotspots)

- Dynamic behavior:

  - unlike AHM, CHM does not rely only on a link-based model

  - represented and managed by an event-based mechanism
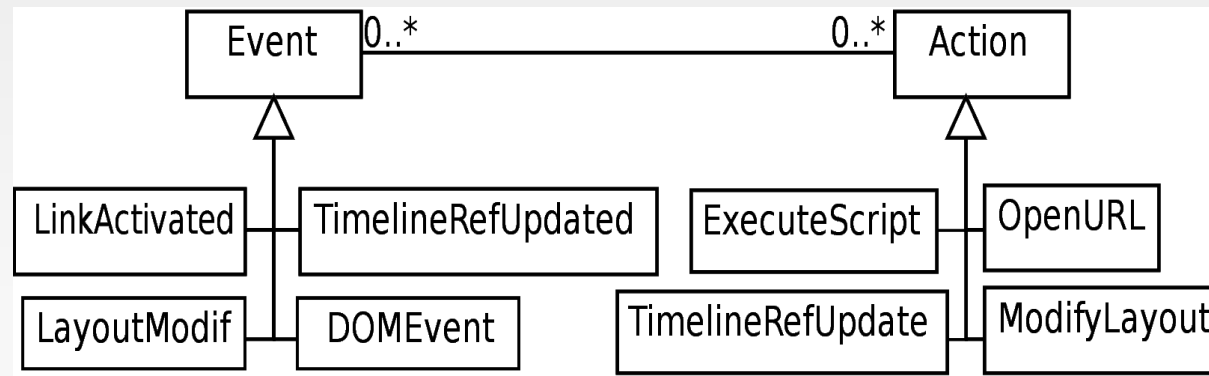
# Architecture Implementation: WebCHM prototype

# Rationale

- WebCHM: practically demonstrates the model
- Public Web-based prototype: http://advene.org/chm/
- Declarative syntax + set of JavaScript libraries
- AV content: HTML5 video (Flash fallback)
- CHM spatial model: HTML layout model + CSS
- CHM temporal model:
  - Add time to Web docs? no established standard way
    - → Timesheets.js (JS implementation of SMIL Timesheets)

# Web-based Hypervideo Syntax

- CHM syntax: extension above the HTML language as namespaced attributes

- A JS-based transformation interprets CHM attributes and generates HTML5 code

- Complex hypervideos are authored as standard Web documents.

    - Common Web content → standard HTML

    - Hypervideo components → HTML + CHM attributes

# Simple Example

```
<!-- Data reader -->
<div chm:component="jsonreader" id="data"  chm:src="data.json" />
<!-- Video Player (renders the TimedMedia) -->
<div chm:component="videoplayer" id="tm"    chm:src="video.ogv"/>
<!-- Table of Content -->
<div chm:component="toc" chm:src="data"     title="Story parts"
    chm:filter="type=='Parts'"     chm:content="${content}"
    chm:timelineref="tm" />
<!-- Captions  -->
<div chm:component="caption" chm:src="data"
    chm:filter="type=='Transcript'" chm:content="${content}"
    chm:timelineref="tm" />
```

# Complete example: nosferatu
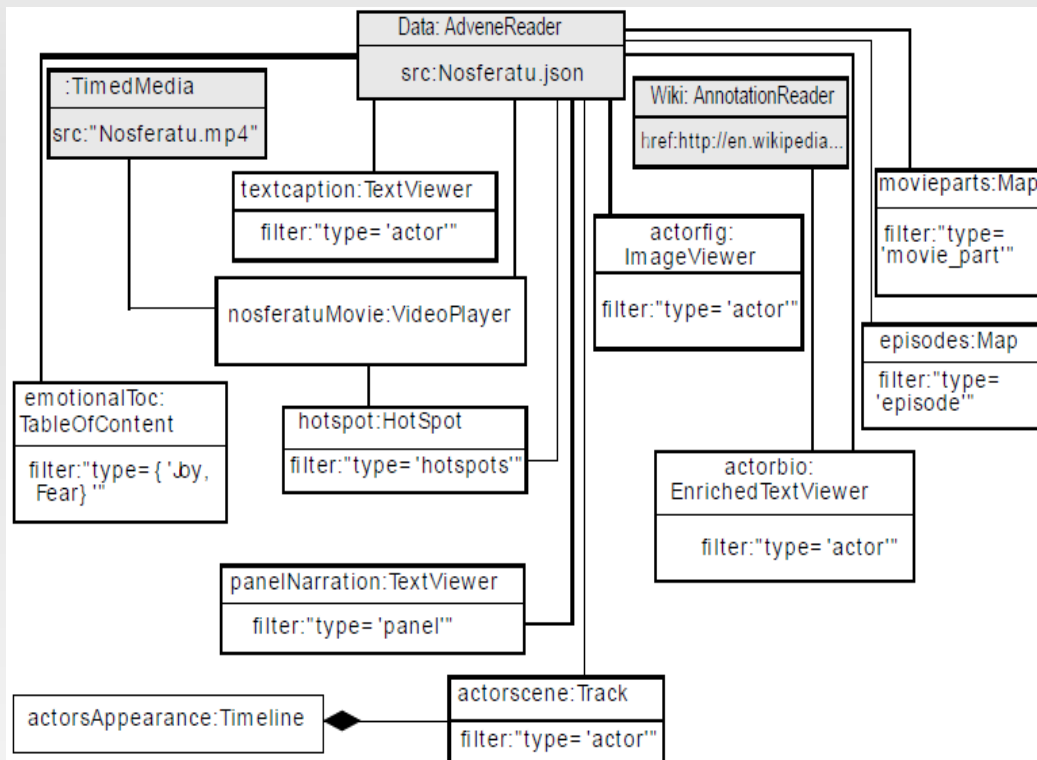


Formal Representation of the Example



Screenshot of the Example Rendering

http://advene.org/chm

# Conclusion

- Further developments are underway

- We aim to enhance CHM and implementation:

  - More meaningful components

  - Study hypervideo perception

  - GUI to ease the authoring

  - Advanced features to enrich interaction possibilities like offering additional mechanisms to end-users

# Conclusion

- The CHM Project Homepage, API and Samples:

    http://www.advene.org/chm

- RSS:

    http://www.advene.org/chm/feed.xml

- Twitter:

    http://twitter.com/chmproject

- Contact:

    msadallah@mail.cerist.dz

    olivier.aubert@liris.cnrs.fr

    yprie@liris.cnrs.fr

# Thank You!