A Hypervideo Model for Learning Objects

Antonio José G. Busson
Federal University of Maranhão
São Luis, MA, Brazil
busson@laws.deinf.ufma.br

André Luiz de B. Damasceno
PUC-Rio
Rio de Janeiro, RJ, Brazil
andre@telemidia.puc-rio.br

Roberto G. de A. Azevedo
PUC-Rio
Rio de Janeiro, RJ, Brazil
razevedo@inf.puc-rio.br

Carlos de Salles Soares Neto
Federal University of Maranhão
São Luis, MA, Brazil
csalles@deinf.ufma.br

Thacyla de Sousa Lima
Federal University of Maranhão
São Luis, MA, Brazil
thacyla@laws.deinf.ufma.br

Sérgio Colcher
PUC-Rio
Rio de Janeiro, RJ, Brazil
colcher@inf.puc-rio.br

ABSTRACT

Learning Objects (LOs) are entities that can be used, reused, or referred during the teaching process. They are commonly embedded into documents that establish spatial and temporal relationships on their contents. Hypervideos LOs allow students to individualize their learning experience with non-linear browsing mechanisms and content adaptation. This paper presents a survey of features for a set of documents representing such LOs as well as desirable aspects that should be expressed during the authoring phase. Also, this paper presents a conceptual model that fits such requirements. The model is implemented by SceneSync, a domain specific language focused on the synchronization and temporal behavior of LOs. As a result of the work, we present a set of LOs specified in SceneSync and a discussion about the identified features, which confirm the expressiveness and applicability of the model.

CCS CONCEPTS

•**Human-centered computing → Hypertext / hypermedia;**

KEYWORDS

Hypervideos; Learning Objects; SceneSync

1 INTRODUCTION

Learning Objects (LOs) are digital entities that can be used, reused, or referred during the learning process with technological support [1, 2]. LOs can be seen as multimedia documents composed of different synchronized media objects, such as images, texts, and videos. A specific, and nowadays popular, type of LO is the hypervideo-based LO, or *hypervideo LO*. Hypervideos are non-linear videos whose playback can be influenced by hyperlinks embedded in elements contained within the video stream [3, 4]. Using these additional elements thus one can specify complex behaviors to LOs such as playing video lectures in sequence, synchronizing lecture videos and additional elements through temporal and spatiotemporal links, and providing different user navigation modes over video lectures [5].

Although general-purpose multimedia authoring languages (such as HTML, SMIL, and NCL) have been used to specify hypervideo LOs (more details on Section 4), there are some advantages on defining a domain-specific language (DSL [6]) for such a purpose. A DSL offers a higher abstraction level to authors, closer to its application domain. This way, LOs can be created similar to how the teacher structures its lectures. Also, DSLs are small languages that do not have as many elements as general purpose languages so that the complexity in the development of authoring tools and exhibition players (or formatters [7]) is usually less time consuming.

This paper presents SceneSync, a DSL for authoring hypervideo LOs. SceneSync is based on a concise and straightforward conceptual model that structures LOs in scenes and provides simple declarative concepts for defining the temporal behavior of LOs. SceneSync was created specifically for the domain of LOs authoring. Therefore, it has a restricted and specific semantics. SceneSync documents represent the concept of interoperable learning objects [8], as the same document can be used to run on both web browsers or iDTV (Interactive Digital Television) plataforms.

The temporal features of SceneSync documents are specified by a timeline-based synchronization model (instead of a structure-based [9] one) that can be easily used by non-programmers, and that can be coupled to different authoring tools. This project decision simplifies the implementation of graphical representation for the SceneSync document model in authoring tools. It also eases the implementation of formatters that maintain the temporal relationships during the document exhibition.

The remainder of this paper is structured as follows. Section 2 presents a study to identify the requirements observed in different hypervideo LOs. Section 3 presents hypermedia authoring languages that are currently used to develop hypermedia LOs. Section 4 contains the core of the proposal, with the construction of a model for hypervideo LOs. The viability of applying SceneSync is discussed in Section 5. Finally, Section 6 brings our final considerations and discusses the results and future work.

2 REQUIREMENTS FOR MODELING HYPERVIDEO-BASED LEARNING OBJECTS

Table 1 summarizes the functional and non-functional requirements we have gathered for the creation of hypervideo-based LOs. These

HT'17, July 4-7, 2017, Prague, Czech Republic.

DOI: http://dx.doi.org/10.1145/3078714.3078739

requirements were gathered through design techniques performed with stakeholders (as detailed in [10]) and by surveying the LO literature for works that have identified similar requirements or preferential features of users when using hypervideo LOs. As discussed in the next sections, those requirements were used as the basis for the creation of the SceneSync conceptual model.

Interactivity and multimedia content have long been seen as ways of supporting and enhance the teaching-learning process. For instance, Vieira *et al.* [11] highlight the importance of the students' interaction and the promotion of their participation in e-learning environments that use videos. Vieira *et al.*'s work compares different e-learning environment patterns and identifies fundamental requirements that should be taken into account when developing such environments. The analyzed patterns were: *Lecture Capture* (recording of a class or lecture); *Talking Head* (top recording of the instructor talking to the camera); *Voice Over Presentation* (slides presentation complemented with narration); and *Interactive Video* (video presentation enriched with multimedia content and interaction features). The main identified requirements were: the usage of short videos, soft transitions, hyperlinks support, and content summarization (for selective viewers).

Zhang *et al.* [12] and Dotta *et al.* [13] studied the preferences of the students and their learning satisfaction on e-learning environments. Zhang *et al.* [12] analyzed four different environments. The first environment uses interactive videos. The second one uses non-interactive videos. The third one does not use any video. And, the fourth environment is a traditional classroom. They concluded that students are significantly better satisfied and they have better learning performances in the first environment, i.e., the one that uses interactive videos. Based on Zhan *et al.* studies, interactivity is an important way to improve learning effectiveness in e-learning environments. Indeed, by providing an individualized control for content access through an organized content index, the student can learn at his pace. Dotta *et al.* [13] surveyed the predilection of students for the use of animations in videos. They found that, according to the majority of the students, the use of animations eases the learning process.

Brecht [16] measured the benefits of using video lectures to complement face-to-face classes. For that, he analyzed three different LO designs. The first design is characterized by the lack of attention to relief and change-of-pace elements (this is similar to the previously mentioned *Lecture Capture* from [11]). The second one included graphics/cartoons and sound/music clips, which were used to provide relief from fatigue during the class. Finally, the third design uses a significantly reduced number of graphics and sounds that were subtly presented in a way that they did not call too much of the students' attention. By comparing those designs, Brecht found that the second design (which uses graphics/cartoons and sound/music clips) had presented the best performance (i.e., the best learning rate and the lowest dropout rate of the students) in comparison to the other two.

Mujacic *et al.* [5] evaluate the performance of students in e-learning environments using hypervideo LOs. For this purpose, they analyzed two groups of students. The first group undertook the course using the traditional model of lectures, while the second group used the hypervideo LOs. By interviewing the students, it was verified that the introduction of the hypervideo LOs, provided for the second group, gave them better insight into the activities, and enabled a higher level of interaction and control over the contents. The LO model features used by Mujacic *et al.* include scenes composition, timeline-based synchronism, and spatiotemporal links.

Table 1: Requirements for a hypervideo model for designing LOs.

FR# - Functional requirements
FR1 - Use multiple short videos instead of a single video (split into scenes) [5, 10, 11];
FR2 - Support for internal and external hyperlinks [10–12, 14, 15];
FR3 - Additional information: video enriched with images, texts, animations, other videos, audios, etc. [10, 13–17];
FR4 - Clickable areas, button panels, questions or quizzes. [10–12] [14, 15];
FR5 - Non-linear structure: no predefined course of playback, but a graph-structure of scenes [10–12, 14];
FR6- Timeline-based synchronism paradigm: media items are placed along a time axis, but possibly on different tracks [5, 10];
FR7 - Temporal links: jump labels on a timeline of a scene [5, 10].
NF# - Non-functional requirements
NF1 - Transition effects: presentation effect used when the formatter starts or finishes displaying a media [11];
NF2 - Interoperability: run on the Web, Mobile, iDTV [10, 15, 18];
NF3 - Adaptability: adaptation to different contexts [12, 15];
NF4 - Accessibility [15];
NF5 - Durability: resist technological changes without the need for recoding [15];
NF6 - Reusability: reuse of components [10, 15];
NF7 - Reliability: reliable and without errors [10].

Interactivity and multimedia content are also important for including impaired children and improving their learning experience. Focused on mathematics and pre-calculus, Munoz-Soto *et al.* [14] present a LO model for children diagnosed with autism spectrum disorders (ASD). By a preliminary validation of the proposed model, Munoz-Soto *et al.* have verified that the ASD children approved the interaction features and the use of multimedia content to add learning value. Becerra *et al.* [17] present a LO for children with language impairments that uses multiple sounds synchronized with other multimedia. The goal of the proposal is to use sound elements to stimulate children's phonological awareness.

Besides the Web, the iDTV is another important platform for diffusing educational applications. The interactive support and the processing capacity of TV sets and set-top boxes, however, does not enable the execution of complex applications [18]. Gadelha *et al.* [15] present OAF-TV, a scheme for the description of functional LOs for iDTV that can be adapted and reused to meet different teaching contexts and methodologies. OAF-TV is defined as a digital artifact that can have components to be reused in different learning contexts supporting interactivity, which allows the provision of exclusive content tuned to specific user needs and profiles.

3 HYPERMEDIA AUTHORING LANGUAGES

It is possible to develop hypervideo LOs using declarative languages such as SMIL, HTML, XMT, and NCL. However, those languages are designed for general purpose hypermedia applications and do not implement specific features of the educational domain.

SMIL (Synchronized Multimedia Integration Language) [19] is an XML-based declarative language for the specification of interactive multimedia presentations for the Web. SMIL provides temporal behavior constructions, and it allows authors to associate hyperlinks to media objects and to specify spatial layouts. The SMIL specification is organized into modules [20], which makes it possible to reuse parts of the language. However, in April 2012, W3C has extinguished the committee responsible for the language evolution, defining HTML5 [21] as a related topic [22].

HTML (together with some package format, such as SCORM [23], which will be discussed in the next section) is currently the most used language for developing LOs. Using HTML has the big advantages of being supported on many platforms. HTML5 provided multimedia features such as <video> and <audio> elements, which were a big advancement in the language towards a more multimedia environment. However, even today, the main features of LOs using HTML are usually developed using Javascript. Differently, we are more interested in high-level declarative features for hypervideo LOs. As discussed later in this paper, to take advantage of the high-availability of HTML5 the SceneSync model can also be converted to it (plus some Javascript code).

XMT (eXtensible MPEG-4 Textual Format) [24] is the MPEG-4 standard for interactive multimedia content representation. XMT has two representation levels: XMT-A and XMT-O. XMT-A is a direct textual (XML-based) representation of the MPEG-4 binary format used for transmission, named BIFS (BInary Format for Scenes) [25]. This way, XMT-A has the same expressiveness of BIFS. XMT-O, on the other hand, is an attempt of simplifying the authoring process, by providing higher-level language (based on SMIL elements). The XMT-O elements are temporarily arranged and synchronized with the SMIL time containers (<par>, <seq>, and <excl>). This simple set of temporal compositions hinders the definition of complex temporal relationships. Thus, in such cases, it is necessary to define compositions with several nested levels. Moreover, those compositions force authors to structure the document according to the temporal presentation specification [26]. Some modules of SMIL and XMT also incorporates characteristics from the X3D language [27], which allows content interchange between XMT, SMIL and X3D users, respecting the compatibility limit between the languages.

NCL (Nested Context Language) [28] is a declarative language for the authoring of hypermedia documents based on the NCM (Nested Context Model) [29] conceptual model. Initially, the language was designed for the Web, but today its main usage is as the declarative language of the Brazilian Digital Television System (SBTVD) and the H.761 recommendation of International Telecommunications Union for IPTV services [30]. In both cases, NCL supports the development of interactive multimedia applications for Digital TV. Similar to SMIL and XMT, NCL is also designed as a set of modules, which can be combined to produce different language profiles. The modules can also be reused by other languages. The latest NCL version, 3.0 [31], reviewed some features contained in older versions. In special, the new version introduces two new features: navigation using keys and support for animations. Also, it introduces changes in the composite node template functionality and restructures the specification of hypermedia connectors in order to have a more concise notation. NCL, however, has several elements that are used to facilitate the reuse structuring of certain elements, and which does not have a direct influence on the application's semantic [32]. Hypervideo LOs can also be specified with NCL, but NCM is a broad-spectrum model so that teachers must rationalize all the complexity of the model to be able to define new LOs.

In contrast to the general purpose hypermedia languages above, this paper proposes a declarative language to specify hypervideo documents in the educational domain. Moreover, SceneSync does not take into account human aspects such as expressiveness and usability, which would be essential when designing multimedia authoring tools. Indeed, SceneSync is not designed to be used directly by end users, but to be used by authoring tools and players, or converted to one of the languages discussed in this section.

4 THE SCENESYNC MODEL

This section presents the *SceneSync Model* (SSM), a model for creating hypervideo LOs. SSM has been designed taking into account the requirements described in Section 2. Figure 1 shows the model entities.

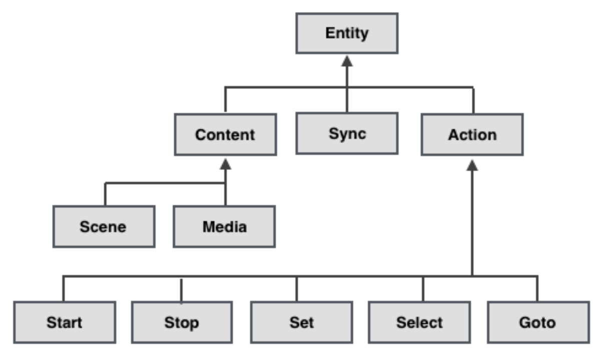

Figure 1: Class hierarchy of the SSM entities.

A *Content* entity can be a *Scene* or a *Media object*. The *Scene* entity is used as an abstraction for representing multimedia compositions (FR1 and FR3 requirements). A *Scene* entity instance has, as its content, a collection of *Media objects* and *Synchronization anchors* (*Sync*). For simplifying the LOs modeling, a *Scene* entity cannot contain others scene objects, i.e., SSM does not allow nesting compositions.

The timeline abstraction is used for providing synchronism among *Media objects* (FR6 requirement) in the same *Scene*. Every *Content* object (*Scene* or *Media*) has an internal clock that can be used as a reference for defining temporal anchors on the scene. An anchor is defined by the *Sync* entity, and has, as its content, a collection of *Action* objects.

An *Action* defines the relationships between the *Anchor* and *Content* objects, and can be of one of the following five types: *Start*, which starts the presentation of a media object; *Stop*, which stops the presentation of a media object; *Set*, which sets the properties of a media object; and *Select* and *Goto*, which redirect the presentation to another time point or scene node. The difference between the *Select* and *Goto* actions is that the *Select* action is triggered via the user interaction (requirements FR2, FR4 and FR7), whereas the *Goto* action is triggered when the scene reaches a specific time point. In special, these actions enable modeling non-linear LOs (FR5 requirement).

4.1 The SceneSync Language

LOs specified in SceneSync are compliant with the SCORM (Sharable Content Object Reference Model) [23] standard specifications, as they can be encapsulated into a Shareable Content Object (SCO) package (Figure 2). Besides the SceneSync document (c), an SCO package should contain three additional files: a manifest file SCORM (a), which informs AVA systems how, when, and where the SceneSync document should be executed; a LOM [33] metadata file (b), used to facilitate the searching of LOs; and the media files (d).

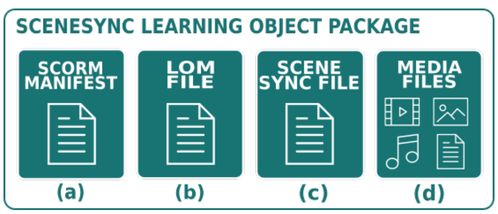

Figure 2: SceneSync Learning Objects Package

The SSM model is instantiated in XML by the SceneSync language. Similar to SMIL and NCL, the SceneSync language is specified using a modular approach in XML Schema [34]. Table 2 details the three modules of the language: *Structural*, *Content*, and *Synchronization.*

4.1.1 Structure module.

The structure module defines three main elements: <scenesync>, <head>, and <body>. The <scenesync> is the root element of the language, and it has the <head> and <body> elements as children. It also has the *id* and *xmls* attributes, which identify the application and the standard scheme, respectively. The <head> element may have an optional <meta> element that allows authors to specify metadata about the document. The <body> element has as children the elements that describe the presentation content, such as media and synchronism objects.

4.1.2 Content module.

A SceneSync presentation is composed of one or more scene nodes, represented by the <scene> element. Besides the identifier, a <scene> has a collection of media objects and synchronization elements as children. Each media object type is represented by a different first class element with their proper attributes: <image>, <text>, <audio>, or <video>. Media objects have as attributes: *id*, a unique identifier of the media object; *src*, which defines the media object contents *URI*; and others attributes that define presentation characteristics, such as *top*, *left*, *width*, *height*, and *transparency*. Media objects also contain a list of synchronism objects (<sync> elements, discussed next).

4.1.3 Synchronization module.

The synchronization among the media objects and scenes are defined by the <sync> element. The <sync> element has the *id* and *time* attributes. The *id* attribute identifies a <sync> element unambiguously in the document. The *time* attribute defines a temporal anchor in its parent element timeline (a <scene> or a media object). The children actions of the <sync> element will be fired when the parent element presentation reaches the time specified by the *time* attribute.

In the SceneSync language, every possible action over a <scene> or media object is represented by a different element. The <start> and <stop> elements represent actions that, respectively, starts or stops the presentation of a <scene> or media object. The <set> element represents an action that changes the property of a media object. The <goto> and <select> elements represent actions to redirect the multimedia presentation to another moment in time or to another <scene>. The difference between the <goto> and <select> actions is that the former is always triggered when the presentation reaches the specified time, whereas the latter is only triggered by a user interaction (such as a key press or mouse click).

Besides their identifiers, action objects have also the *target* attribute, which defines the target object of the action. In particular, the <set> element has a list of <property> elements. A <property> element specifies the property that will be modified (*name* attribute) and the new value to be defined (*value* attribute). The <select> element has the *key* attribute, which defines a key that triggers the action. The *timeEvent* attribute, which may be specified in a <select> and <goto>, defines the time anchor the scene node will be redirected to.

4.2 Modeling a LO using SceneSync

To help the understanding of the SceneSync language, this subsection presents a step-by-step development of a simple hypervideo LO, the "Sorting Algorithms" LO. This LO begins with the video of a teacher introducing general concepts related to sorting algorithms. At some point during his lecture, the teacher asks the viewer to choose whether he wants to learn more about the *InsertSort* or the *QuickSort* sorting algorithm. The LO then presents a specific lecture (about *InsertSort* or *QuickSort*) based on the user choice.

To implement the "Sorting Algorithms" LO, we divide it into three <scene> elements:

(1) *IntroductoryScene*, which contains:
   (a) one introductory video about sorting algorithms;
   (b) one image that represents the *InsertSort* option; and
   (c) one image that represents the *QuickSort* option.
(2) *InsertSortScene*, which contains:
   (a) one background image;
   (b) one video about the *InsertSort* algorithm; and
   (c) one image that illustrates an example of the *InsertSort* algorithm.
(3) *QuickSortScene*, which contains:
   (a) one background image;
   (b) one video about the *QuickSort* algorithm; and
   (c) one image that illustrates an example of the *QuickSort* algorithm.

Table 2: SceneSync language modules, elements, and attributes.

Element	Attributes	Child Elements
	Structure module	
scenesync	id, xmls	(head?, body)
head	-	meta?
body	-	(scene - media - sync)*
	Content module	
scene	id	(image, text, audio, video, sync)*
audio	id, src, volume	sync*
image	id, src, left, top, width, height, transparency, layer	sync*
video	id, src, left, top, width, height, transparency, volume, layer	sync*
text	id, src, left, top, style, align, color, fontfamily, fontsize, transparency, layer	sync*
	Synchronization module	
sync	id, time	(start, stop, set, select, goto)*
start	id, target	-
stop	id, target	-
set	id, target	property*
goto	id, target, timeevent	-
select	id, target, timeevent, key	-
property	id, name, value	-

Listing 1 shows part of the "Sorting algorithm" source code, highlighting the <scene> and media objects, organized as above. Next, we discuss the behavior specification of each of the <scene>s.

The *IntroductoryScene* starts with the "vid_alg" video. This can be modeled with a <sync> element containing a child object of action <start> pointing to a video object (as shown in lines 3–5 of Listing 2). Note that we do not need to provide the value of the *time* attribute, which by default is 0.

Moreover, when the teacher invites the viewer to choose the next lecture (*InsertSortScene* or *QuickSortScene*) two images are displayed, illustrating those options to the viewer. It is also necessary to enable the user interactions that will lead to the selection of the respective scenes. To do this, we create a new <sync> element (lines 6–11 of Listing 2) with the *time* attribute set to "45s" (the time the teacher asks the user to interact). Inside this <sync> we define two <start> actions, the first one created to start the "img_insert" object, and the second one created to start the "img_quick" object. To enable user interaction, we define two <select> actions, which enable the key interaction feature.

Finally, when the main video of the *IntroductoryScene* ends (65s), the scene presentation should go back to 45s, and stay in a loop until the user chooses an option. We use the <goto> element to implement such behavior (lines 12–14 of Listing 2). The <goto> element is used with *target* and *timeEvent* attributes set as "IntroductoryScene" and "45s", respectively. Figure 3 shows the preview and temporal view of the *IntroductoryScene*.

Listing 1: Scenes and media objects of the "Sorting Algorithm" LO.

```
<scene id="IntroductoryScene">
  <video id="vid_alg" src="intro.mp4"
      width="100%" height="100%" />
  <image id="img_insert" src="img1.png"
      left="30%" top="80%" width="30%"
      height="20%" />
  <image id="img_quick" src="img2.png" left="65%"
      top="80%" width="30%" height="20%" />
  ...
</scene>
<scene id="InsertSortScene">
  <video id="vid_insert" src="video1.mp4"
      width="100%" height="100%" />
  <image id="back_insert" src="back.png"
      width="100%" height="100%" />
  <image id="img_alg_insert" src="img2.png"
      left="65%" width="80%" width="30%"
      height="20%" />
  ...
</scene>
<scene id="QuickSortScene">
  <video id="vid_quick" src="video1.mp4"
      width="100%" height="100%" />
  <image id="back_quick" src="back.png"
      width="100%" height="100%" />
  <image id="img_alg_quick" src="img2.png"
      left="65%" width="80%" width="30%"
      height="20%" />
  ...
</scene>
```

Listing 2: Source code defining the temporal behavior of *IntroductoryScene* of the "Sorting Algorithm" LO.

```
<scene id="IntroductoryScene">
...
 <sync>
  <start target="vid_intro" />
 </sync>
 <sync time="45">
  <start target="img_insert" />
  <start target="img_quick" />
  <select key="1" target="InsertSortScene"/>
  <select key="2" target="QuickSortScene"/>
 </sync>
 <sync time="65s">
  <goto target="IntroductoryScene"
       timeEvent="45s"/>
 </sync>
</scene>
```

Listing 3: Source code defining the temporal behavior of *InsertSortScene* of the "Sorting Algorithm" LO.

```
1  <scene id="InsertSortScene">
2  ...
3   <sync>
4    <start target="vid_insert" />
5   </sync>
6   <sync time="5s">
7    <start target="img_back_insert" />
8    <start target="img_alg_insert" />
9    <set target="vid_insert">
10    <property name="left" value="5%" />
11    <property name="top" value="5%" />
12    <property name="width" value="17%" />
13    <property name="height" value="10%" />
14   </set>
15   </sync>
16 </scene>
```

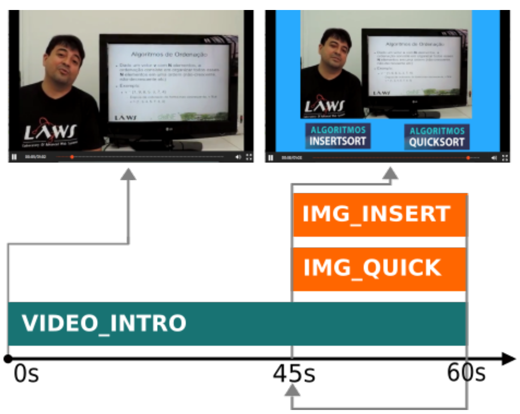

Figure 3: Preview (top) and temporal view (bottom) of *IntroductoryScene.*

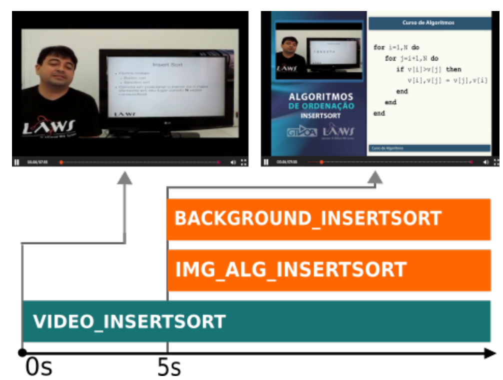

Figure 4: Preview (top) and temporal view (bottom) of *InsertSortScene.*

The second (*InsertSortScene*) and the third (*QuickSortScene*) scenes have a similar behavior. Therefore, we will discuss only the second scene here. The *InsertSortScene* starts by displaying the video about the *InsertSort* algorithm. For that, a <sync> element is defined containing a <start> action referring the video presentation "vid_insert" (Listing 3, lines 3–5). As an additional behavior of this scene, at 5 seconds: (1) the main video must be resized and moved to the upper left corner; (2) a background image must be started (behind the video and occupying the entire scene); and (3) the image containing an example of the *InsertSort* algorithm must be displayed on the right side of the screen. To implement that, we used the <sync> element defined in Listing 3, lines 6–15. Figure 4 shows the preview and temporal view of the *InsertSortScene*.

5 ANALYSIS

In order to show the viability of using the SceneSync model we have asked for 142 stakeholders (teachers and educational content creators) to develop LOs using the concepts provided by SceneSync. To allow them to use the SceneSync concepts and create an executable LO, we have used the Cacuriá [35], which is an authoring tool that graphically implements the SSM concepts. The LOs were created as part of 7 workshops and 1 technical session held in 5 Brazilian cities during three years of a research project. A complete description of the evaluation of the Cacuriá authoring tool can be found in [36]. Here, we summarize and analyze those results with regards to the underlying SceneSync model.

Both in the workshops and in the technical session, first, the participants were presented to the process of authoring LOs using SceneSync concepts and the Cacuriá authoring tool. Then, they were asked to develop a LO about some topic of their interest (in the workshops) or a predefined LO (in the technical session). In total, 32 LOs [1] (Figure 5) were specified in SceneSync using the Cacuriá tool.

Table 3 shows the developed LOs, with their respective disciplines and identified functional requirements. As can be seen, all the functional requirements that guided us in the development of SceneSync could be successfully explored by the authors of LOs.

All the LO authors were able to use the following features successfully: multimedia compositions (FR1) through the <scene> element; additional video information such as images, texts, animations, another videos, and audio (FR3); timeline-based synchronism (FR6), through the <sync> and action elements; clickable areas (FR4) and non-linear narrative structures (FR5), through the <sync> and <select> elements; temporal links, spatial-temporal links (FR7), <sync>, <goto>, and <select> elements.

Cacuriá supported the external hyperlinks (FR2) feature only in the last workshop. Thus, only three LOs ("05-Citation", "08-Scientific Work", and "15 - Game of Thrones quiz") have used it. Probably, if that functionality had been provided before, more LOs would have been created using this feature.

6 CONCLUSION

This paper presents the SceneSync model (SSM), a hypervideo model for LOs, and its instantiation as an XML-based language. Compared to other approaches to implementing hypervideo LOs, such as using a general-purpose multimedia languages directly, SSM provides abstractions close to the hypervideo LOs domain, so that it simplifies the authoring process (which is now in a proper abstraction level), the development of authoring tools on a similar domain (such as the Cacuriá authoring tool), and the development of players for hypervideo LOs. Moreover, the final artifact created using SSM can be converted to other lower-level languages, such as HTML (to run on the Web) to NCL (to run in iDTV platforms). Indeed, the Cacuriá authoring tool can export to both of those formats.

In the process of developing SSM, an additional contribution of the paper is the set of requirements defined in Section 2. Aiming at characterizing the hypervideo LOs domain, we present a survey of features for hypervideo LOs documents gathered through design techniques performed with stakeholders and from a literature review, focused on functional features and users' preferences when using hypervideo LOs. The SSM supports all the gathered requirements. Also, to show the viability of our proposal, we analyzed 32 LOs created by 142 stakeholders, showing that most of the features were indeed used by them when creating hypervideo LOs.

Based on the results presented in Section 5 we have good confidence that SSM is useful for a broad range of hypervideo LOs. So far, however, SSM was not designed to be directly used by teachers. In future work, we plan to perform a qualitative study of the communicability and usability of the model without the interposing of a visual authoring tool. We also want to investigate the reusability degree that we can achieve when using SSM together with the SCORM standard.

Still focused on reusability, we plan to extend the SSM features by providing native support for defining new concepts in the model through templates. Templates can improve the reuse of the specification of LOs and help to create a visual identity among sets of LOs.

Finally, another future work is related to the identification and analysis of design patterns that could be useful in the scope of hypervideo LOs. In such a future work, the focus is on finding recurrent code structures and best practices that authors can reuse when creating new LOs. Researching design patterns can be useful in at least two ways: for defining authoring guidelines and ii) for providing new concepts, which can be used to extend SSM.

[1] Available at https://goo.gl/I6SyKu

ACKNOWLEDGMENTS

This work was supported by RNP, the National Research and Educational Network from Brazil, and was developed in the context of the RNP Working Groups program, during the cycles 2012–2013, 2013–2014, and 2015. The authors also thank CAPES, FAPEMA, and CNPq for their additional support.

REFERENCES

[1] IEEE Learning Technology Standards Committee et al. Draft standard for learning object metadata. *Accessed July*, 14:2002, 2002.

[2] David A Wiley. *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy.* 2003.

[3] Britta Meixner, Katarzyna Matusik, Christoph Grill, and Harald Kosch. Towards an easy to use authoring tool for interactive non-linear video. *Multimedia Tools and Applications*, 70(2):1251–1276, 2014.

[4] Nitin Sawhney, David Balcom, and Ian Smith. Hypercafe: narrative and aesthetic properties of hypervideo. In *Proceedings of the the seventh ACM conference on Hypertext*, pages 1–10. ACM, 1996.

[5] Samra Mujacic, Matjaz Debevc, Primoz Kosec, Marcus Bloice, and Andreas Holzinger. Modeling, design, development and evaluation of a hypervideo presentation for digital systems teaching and learning. *Multimedia Tools and Applications*, 58(2):435–452, 2012.

[6] Marjan Mernik, Jan Heering, and Anthony M Sloane. When and how to develop domain-specific languages. *ACM Computing Surveys (CSUR)*, pages 316–344, 2005.

[7] Luiz Fernando G Soares, Rogério F Rodrigues, and Débora C Muchaluat Saade. Modeling, authoring and formatting hypermedia documents in the hyperprop system. *Multimedia Systems*, 8(2):118–134, 2000.

[8] Norm Friesen. Interoperability and learning objects: An overview of e-learning standardization. *Interdisciplinary Journal of Knowledge and Learning Objects*, 1 (1):23–31, 2005.

[9] Lynda Hardman, Guido Van Rossum, and Dick CA Bulterman. Structured multimedia authoring. In *Proceedings of the first ACM international conference on Multimedia*, pages 283–289. ACM, 1993.

[10] André Luiz de Brandão Damasceno, Carlos Salles Soares Neto, and Simone Diniz Junqueira Barbosa. Integrating participatory and interaction design of an authoring tool for learning objects involving a multidisciplinary team. In *Proceedings of the International Conference on Human-Computer Interaction*, 2017.

[11] Isabel Vieira, Ana Paula Lopes, and Filomena Soares. The potential benefits of using videos in higher education. In *Proceedings of EDULEARN14 Conference*, pages 0750–0756. IATED Publications, 2014.

[12] Dongsong Zhang, Lina Zhou, Robert O Briggs, and Jay F Nunamaker. Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness. *Information & management*, 43(1):15–27, 2006.

[13] Silvia C Dotta, Erica FC Jorge, Edson P Pimentel, and Juliana C Braga. Análise das preferências dos estudantes no uso de videoaulas: Uma experiência na educação a distância. In *Anais do Workshop de Informática na Escola*, volume 1, page 21, 2013.

[14] Roberto Munoz-Soto, Carlos Becerra, René Noël, Thiago Barcelos, Rodolfo Villarroel, Sandra Kreisel, and Matías Camblor. Proyect@ matemáticas: A learning object for supporting the practitioners in autism spectrum disorders. In *Learning Objects and Technology (LACLO), Latin American Conference on*, pages 1–6. IEEE, 2016.

[15] Bruno Freitas Gadelha, Alberto Nogueira CASTRO-JR, and Hugo Fuks. Representando objetos de aprendizagem funcionais para tvdi. In *SET2007–Congresso da Sociedade Brasileira de Engenharia de Televisão, São Paulo*, 2007.

[16] H David Brecht. Learning from online video lectures. *Journal of Information Technology Education*, 11:227–250, 2012.

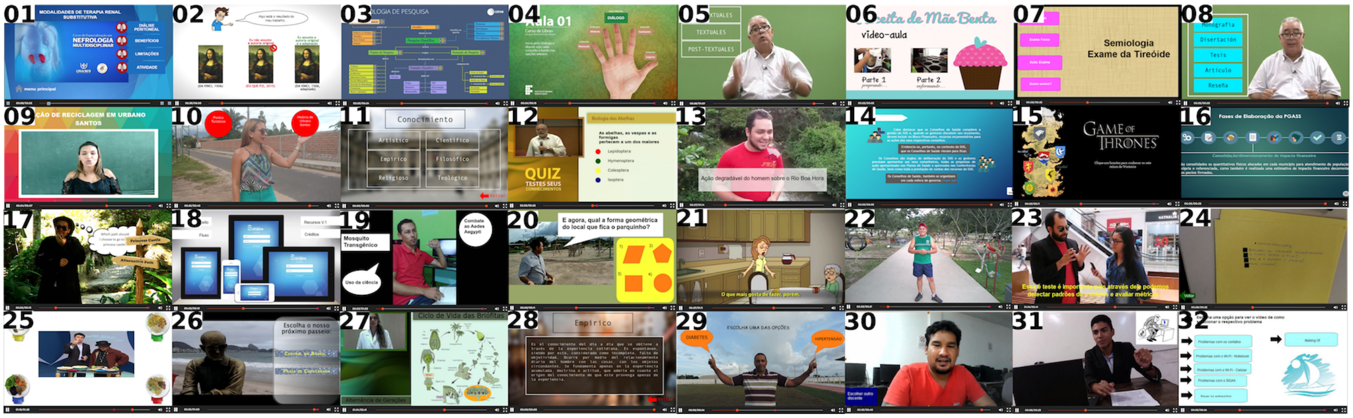

Figure 5: LOs developed by stakeholders.

Table 3: Functional requirements used by every LO‘s authors

#	Title	Discipline	FR1	FR2	FR3	FR4	FR5	FR6	FR7
01	Nephrology	medicine	Yes	No	Yes	Yes	Yes	Yes	Yes
02	Plagiarism	education	Yes	No	Yes	Yes	Yes	Yes	Yes
03	Scientific methodology	philosophy	Yes	No	Yes	Yes	Yes	Yes	Yes
04	LIBRAS (Brazilian Sign Language)	accessibility	Yes	No	Yes	Yes	Yes	Yes	Yes
05	Citation	education	Yes	Yes	Yes	Yes	Yes	Yes	Yes
06	Cupcake recipe	culinary	Yes	No	Yes	Yes	Yes	Yes	Yes
07	Thyroid	medicine	Yes	No	Yes	Yes	Yes	Yes	Yes
08	Scientific Work	education	Yes	Yes	Yes	Yes	Yes	Yes	Yes
09	Recycling actions	recycling	Yes	No	Yes	Yes	Yes	Yes	Yes
10	Tourist points of "Urbano Santos"	turism	Yes	No	Yes	Yes	Yes	Yes	Yes
11	Knowledge	philosophy	Yes	No	Yes	Yes	Yes	Yes	Yes
12	Importance of bees	biology	Yes	No	Yes	Yes	Yes	Yes	Yes
13	"Boa Hora" river	geography	Yes	No	Yes	Yes	Yes	Yes	Yes
14	PGASS Elaboration	management	Yes	No	Yes	Yes	Yes	Yes	Yes
15	"Game of Thrones" quiz	entertainment	Yes	Yes	Yes	Yes	Yes	Yes	Yes
16	Social control in SUS	health	Yes	No	Yes	Yes	Yes	Yes	Yes
17	Princess abduction	entertainment	Yes	No	Yes	Yes	Yes	Yes	Yes
18	Sucupira platform	technology	Yes	No	Yes	Yes	Yes	Yes	Yes
19	Aedes aegypti	biology	Yes	No	Yes	Yes	Yes	Yes	Yes
20	Geometry in the square	mathematics	Yes	No	Yes	Yes	Yes	Yes	Yes
21	Elderly health	health	Yes	No	Yes	Yes	Yes	Yes	Yes
22	Sedentary Lifestyle	health	Yes	No	Yes	Yes	Yes	Yes	Yes
23	IHC	design	Yes	No	Yes	Yes	Yes	Yes	Yes
24	Communicability	design	Yes	No	Yes	Yes	Yes	Yes	Yes
25	Dishes	culinary	Yes	No	Yes	Yes	Yes	Yes	Yes
26	Route of the day	turism	Yes	No	Yes	Yes	Yes	Yes	Yes
27	Life cycle of bryophytes	biology	Yes	No	Yes	Yes	Yes	Yes	Yes
28	Empiricism	philosophy	Yes	No	Yes	Yes	Yes	Yes	Yes
29	Physical education	health	Yes	No	Yes	Yes	Yes	Yes	Yes
30	Scientific text	education	Yes	No	Yes	Yes	Yes	Yes	Yes
31	Cybersecurity	technology	Yes	No	Yes	Yes	Yes	Yes	Yes
32	Wifi help	technology	Yes	No	Yes	Yes	Yes	Yes	Yes

[17] Carlos Becerra, René Noel, Roberto Munoz, and Ian Quiroga. Explorando aprendo: Learning object to enhance language development in children with specific language impairment. In *Learning Objects and Technology (LACLO), Latin American Conference on*, pages 1–5. IEEE, 2016.

[18] Júlia Marques Carvalho da Silva and Rosa Maria Vicari. Relacionando a televisão digital interativa com o conceito de objetos de aprendizagem: conceitos, aspectos históricos, e perspectivas. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 1, 2009.

[19] P Vuorimaa, D Bulterman, and P Cesar. Smil timesheets 1.0. *W3C Working Draft*, 2008.

[20] Y Ayers, A Cohen, Dick Bulterman, et al. Synchronized multimedia integration language (smil) 2.0. *W3C Recommendations*, 2001.

[21] Ian Hickson and David Hyatt. Html5: A vocabulary and associated apis for html and xhtml. *W3C Working Draft, May*, 25, 2011.

[22] W3C. The symm wg is closed since 01 april 2012. http://www.w3.org/AudioVideo/. [Acessado em 20/05/2016].

[23] Oliver Bohl, Jörg Scheuhase, Ruth Sengler, and Udo Winand. The sharable content object reference model (scorm)-a critical review. In *Computers in education, 2002. proceedings. international conference on*, pages 950–951. IEEE, 2002.

[24] Michelle Kim, Steve Wood, and Lai-Tee Cheok. Extensible mpeg-4 textual format (xmt). In *Proceedings of the 2000 ACM workshops on Multimedia*, pages 71–74. ACM, 2000.

[25] Julien Signes. Binary format for scene (bifs): Combining mpeg-4 media to build rich multimedia services. In *Electronic Imaging'99*, pages 1506–1517. International Society for Optics and Photonics, 1998.

[26] Débora Christina Muchaluat Saade. *Relações em Linguagens de Autoria Hipermídia: Aumentando Reuso e Expressividade.* PhD thesis, Tese de Doutorado, Departamento de Informática, PUC-Rio, Rio de Janeiro, Brasil, 2003.

[27] Leonard Daly and Don Brutzman. X3d: extensible 3d graphics standard. *IEEE Signal Processing Magazine*, 24(6):130, 2007.

[28] Luiz Fernando Gomes Soares, Marcio Ferreira Moreno, Carlos de Salles Soares Neto, and Marcelo Ferreira Moreno. Ginga-ncl: declarative middleware for multimedia iptv services. *IEEE Communications Magazine*, 48(6):74–81, 2010.

[29] Luiz Fernando Gomes Soares and Rogério Ferreira Rodrigues. Nested context model 3.0: Part 1–ncm core. *Monografias em Ciência da Computação do Departamento de Informática, PUC-Rio*, (18/05), 2005.

[30] ITU-T. Recommendation h.761 - nested context language (ncl) and ginga-ncl for iptv services. ITU-T, Geneva, Switzerland, 2009.

[31] Luiz Fernando Gomes Soares and Rogério Ferreira Rodrigues. Nested context language 3.0 part 8–ncl digital tv profiles. *Monografias em Ciência da Computação do Departamento de Informática da PUC-Rio*, 1200(35):06, 2006.

[32] Luiz Fernando Soares, Guilherme Augusto Lima, and Carlos Soares Neto. Ncl 3.1 enhanced dtv profile. *III Workshop de TV Digital Interativa (WTVDI) - Colocated with ACM WebMedia'10*, 2010.

[33] IEEE Learning Technology Standards Committee et al. Draft standard for learning object metadata (lom) ieee 1484.12. 1, 2002.

[34] XML Schema Part. 0: Primer. *W3C Recommendation*, 2, 2001.

[35] André Luiz de Brandão Damasceno, Rosendy Jess Galabo, and Carlos Salles Soares Neto. Cacuriá: Authoring tool for multimedia learning objects. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*, pages 59–66. ACM, 2014.

[36] André Luiz de Brandão Damasceno, Carlos Salles Soares Neto, and Simone Diniz Junqueira Barbosa. Lessons learned from evaluating an authoring tool for learning objects. In *Proceedings of the International Conference on Human-Computer Interaction*, 2017.