Beyond Multimedia Authoring: On the Need for Mulsemedia Authoring Tools

DOUGLAS PAULO DE MATTOS, MídiaCom Lab, Fluminense Federal University, Brazil
DÉBORA C. MUCHALUAT-SAADE, MídiaCom Lab, Fluminense Federal University, Brazil
GHEORGHITA GHINEA, Computer Science Department, Brunel University London, United Kingdom

The mulsemedia (MulSeMedia - Multiple Sensorial Media) concept has been explored to provide users with new sensations using other senses beyond sight and hearing. The demand for producing such applications has motivated various studies in the mulsemedia authoring phase. To encourage researchers to explore new solutions for enhancing the mulsemedia authoring, this survey paper reviews several mulsemedia authoring tools and proposals for representing sensory effects and their characteristics. The article also outlines a set of desirable features for mulsemedia authoring tools. Additionally, a multimedia background is discussed to support the proposed study in the mulsemedia field. Open challenges and future directions regarding the mulsemedia authoring phase are also discussed.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Information systems → Multimedia content creation**; • **Applied computing → Hypertext / hypermedia creation**; • **Human-centered computing** → *Hypertext / hypermedia*; *Graphical user interfaces*; • **Software and its engineering → Integrated and visual development environments**; *Application specific development environments.*

Additional Key Words and Phrases: mulsemedia authoring, mulsemedia tools, mulsemedia model, sensory effects, multimedia authoring, multimedia tools

ACM Reference Format:
Douglas Paulo de Mattos, Débora C. Muchaluat-Saade, and Gheorghita Ghinea. 20XX. Beyond Multimedia Authoring: On the Need for Mulsemedia Authoring Tools. *J. ACM* XX, X, Article XXX (September 20XX), 31 pages.

1 INTRODUCTION

Multimedia applications contain different types of media content such as video, image, text and audio. That is, those applications provide users with audiovisual content that involves only two of the human senses: sight and hearing. However, the majority of human communication is non-verbal and most of us make use of all five senses (sight, hearing, touch, taste and smell) to comprehend the world around us. Moreover, we can also feel internal body changes, which are called interoceptive capabilities. Those capabilities can be classified in the following categories [45]: equilibrioception, sense of balance; thermoception, sense of heat and cold; proprioception, awareness of the position of our body or part of it; nociception, sense of pain; and interoception, capacity of feeling the internal organs.

Authors' addresses: Douglas Paulo de Mattos, douglas@midiacom.uff.br, MídiaCom Lab, Fluminense Federal University, Niterói, Rio de Janeiro, Brazil; Débora C. Muchaluat-Saade, debora@midiacom.uff.br, MídiaCom Lab, Fluminense Federal University, Niterói, Rio de Janeiro, Brazil; Gheorghita Ghinea, george.ghinea@brunel.ac.uk, Computer Science Department, Brunel University London, London, United Kingdom.

0004-5411/20XX/X-ARTXXX $15.00

In order to provide users with new sensations exploring other senses beyond sight and hearing in interactive multimedia applications, a new concept has been proposed: mulsemedia (MulSeMedia - Multiple Sensorial Media) [43]. Sensory effects from the environment perceived by our sensory organs are transformed into electrical signals that are transmitted to the brain, processed and interpreted to create a perception. The whole process is studied by neuroscience, which discusses essential concepts that should be considered in scientific research efforts in the area of mulsemedia applications.

Sensory effects in multimedia applications have provided users with new immersive content increasing their quality of experience (QoE) [63, 71, 88, 91]. Effects are created using light, wind, aroma etc. As an example of an application with multiple sensory effects, we have 4D cinemas, where motion chairs are synchronized with movies' audiovisual content. Indeed, mulsemedia technology is moving from highly specialised niches to the mainstream, with devices becoming increasingly affordable [73]. One of the most important domains that evidences this transition is that of digital games. In this respect, mulsemedia applications are able to provide an immersive environment to increase the game reality and players' QoE. In the entertainment industry, sensory effects are also applied to simulators (e.g., flight, driving etc.) in order to make the simulation experience more realistic. The application of sensory effects in the healthcare area, specifically in therapeutic use for people with special needs (e.g., learning disabilities, autism, Alzheimer's disease and dementia) is discussed in [43]. All of these applications aim at providing immersive environments to increase the user QoE in multimedia applications. Several studies [21, 31, 46, 56, 73] have been published in the literature discussing solutions that integrate, to varying degrees, sensory effects with multimedia applications.

Concerning the development of mulsemedia applications, this can subdivided in three phases [27]: authoring (or production), distribution, and rendering of applications in the physical environment. In the authoring phase, which represents the focus of this paper, sensory effects can be defined through digital capturing and processing of data obtained from sensors; automatic extraction of sensory effects from audiovisual contents; and manual specification by authors. Furthermore, we can combine manual specification of authors with a crowdsourcing approach. In [29], users give suggestions about time intervals in which specific sensory effects could be activated according to an audiovisual content provided by mulsemedia application authors. This approach allows authors to enhance their specifications of time intervals in which sensory effects should be rendered with audiovisual contents.

Regarding advances in mulsemedia application modeling in the authoring phase, studies in [46, 74] provide sensory effects at a low-level abstraction by identifying sensors and actuators. This approach is frequently used in IoT (Internet of Things) solutions supporting the inherent heterogeneity of devices in IoT environments. However, this approach is not suitable for mulsemedia systems [73] since it does not allow the specification of spatial and temporal behavior of multimedia content and sensory effects at a high-level abstraction as in multimedia model approaches.

Another solution for representing sensory effect metadata is the MPEG-V standard [56], which uses the timeline-based paradigm to synchronize sensory effects with existing multimedia content. Additionally, other proposals focus on easing the specification of sensory effect metadata and the annotation of multimedia content with sensory effects by using graphical tools as in [21, 55, 87]. However, all of these initiatives use the timeline-based paradigm, which has several inherent limitations [14]. Moreover, these solutions do not allow authors to specify an entire mulsemedia application by defining the spatial and temporal behavior of media items and effects. The approach presented in [31], in contrast to solutions based on MPEG-V, allows the specification of an entire mulsemedia application by using an event-based paradigm to temporally synchronize sensory effects and multimedia content. Another approach is that described in [33], which makes use of

templates and wizards. However, this solution makes the tool less expressive by restricting authors to the set of predefined applications available in the tool. Moreover, authors are not able to modify the behavior of these applications.

Investigating the challenge of modeling immersive environments with multiple sensory effects is essential for advancing mulsemedia applications. In [27], the demand for tools that enhance mulsemedia application development is highlighted. Moreover, the authors emphasize the importance of the temporal synchronization of sensory effects to support the effectiveness of mulsemedia applications.

To encourage researchers to explore new solutions for the mulsemedia authoring phase, this paper reviews several mulsemedia authoring tools. We also investigate different mulsemedia models since they not only support the representation of mulsemedia applications but also support the implementation of mulsemedia tools. While offering a review of mulsemedia authoring tools, we also aim at stimulating the production of mulsemedia applications.

The paper also gives a multimedia background to support our study in the mulsemedia field. We cite several non-recent references regarding traditional multimedia modeling and authoring since this field is a classic topic. We aim to bring back basic concepts of multimedia modeling and the multimedia community's contributions concerning the investigation of authoring tools over the years.

The remainder of this survey is organized as follows. Section 2 discusses multimedia modeling and authoring tools. Section 3 discusses mulsemedia modeling. In Section 4, we present mulsemedia authoring tools. Section 5 presents a set of desirable features for mulsemedia authoring tools, which are derived from our study of multimedia background and from mulsemedia tools and models discussed in this survey. We compare mulsemedia authoring tools based on those features in Section 6. Section 7 discusses simulators and players for mulsemedia applications. We conclude our survey in Section 8, highlighting open challenges regarding mulsemedia authoring tools.

2 MULTIMEDIA AUTHORING

This section presents work regarding multimedia modeling and tools for authoring multimedia applications. We discuss the temporal synchronization paradigms that underlie different multimedia conceptual models. We also highlight advantages and disadvantages of those paradigms and give a comparison among them. Afterwards, the section discusses different authoring GUI (graphical user interface) approaches, authoring tools and presents some requirements used for comparing them. Those requirements are also essential to support our discussion about mulsemedia authoring tools in the following sections. We finalize the section giving a comparison table to sum up the characteristics of each of the multimedia authoring tools discussed in this section.

2.1 Modeling: Temporal Syncronization Paradigms

Multimedia (or hypermedia) documents are composed of media items and spatio-temporal relationships among them. These document components are expressed through multimedia conceptual models, whose main entities are represented by nodes, links and composite nodes. Temporal relationships among media items can be specified using different temporal synchronization paradigms. The main paradigms [18, 48, 49, 79, 80] discussed in the multimedia community are: script, timeline, graph, hierarchy/structure, constraint and event-based.

2.1.1 Script-based Paradigm. The script-based paradigm uses imperative programming languages to define the temporal and spatial behavior of multimedia documents. This paradigm is consequently more flexible and expressive than other paradigms. However, it requires programming knowledge from authors. The script paradigm also makes the spatio-temporal visualization harder as it does

not provide a high-level abstraction to deal with the document structure. Videobook [66] and Harmony [40] are examples of tools for authoring hypermedia documents that use multimedia models based on the script paradigm.

In [10], a multimedia synchronization toolkit, called Nsync, is proposed. The toolkit does not provide an authoring tool, but consists of a declarative synchronization definition language and a run-time presentation management system. Although this language is declarative, it uses scripts to specify the temporal layout of multimedia applications.

In the web scenario, HTML5 [25] is also an example of a declarative authoring language using the script paradigm. Although it introduces new elements for improving multimedia content support, mainly video and audio, HTML5 uses ECMAScript code to provide more expressive multimedia documents for the Web.

2.1.2 Timeline-based Paradigm. In respect of the timeline-based paradigm, it is effective when media object duration is known a priori or is explicitly defined, since media objects are directly arranged in a temporal axis in this paradigm. On the other hand, it is not appropriate to represent media object temporal order when multimedia presentations have asynchronous behavior as happens in hypermedia. Conditional synchronization cannot be directly defined among media items either. If any object duration is changed, the author needs to reorganize media items in time in order to keep the specified temporal dependencies. Using the timeline-based paradigm does not allow a formatter (multimedia player or presentation engine) to make runtime adjustments in case of a media network reception delay. In other words, multimedia presentations lose their temporal synchronization when that happens. Besides, content adaptation and timeless compositions cannot be expressed directly in a timeline representation. Despite those limitations, this paradigm is widely used in commercial software and is very simple to be understood by non-expert authors. Subsection 2.3 presents some of these tools.

2.1.3 Graph-based Paradigm. The graph-based paradigm uses formalism for defining document synchronization. This paradigm thus takes advantage of several formal models available in the literature. We identify two dominant formal techniques: the flowchart model [90] and the directed graph model [17]. The latter type is mainly represented by Petri nets [70] and the statechart model [32]. The flowchart model is a chart that describes media items behavior step by step using graphical elements. These elements represent actions and decision points that are linked through arrows in order to specify the presentation control flow. This technique is similar to using imperative programming to define the document behavior, but uses visual elements instead.

Timed Petri nets [70] are bipartite directed graphs that contain the following components: places, transitions and arcs. Each place has tokens and a duration. Arcs are used to connect places and transitions. A transition is fired when all of its input places contain tokens. After a transition is triggered, those tokens are moved to its output places. Tokens then remain blocked until the place duration finishes. HTSPN [89], Trellis [41] and caT [65] are multimedia models and tools that use this synchronization paradigm. This approach has the advantage of providing formal verification of document temporal behavior.

Another type of formal definition based on directed graph models is the statechart-based paradigm, which uses statecharts to specify media items and temporal relations. In this chart, states represent media items and transitions define a state hierarchy. Using transitions, authors can define if media items are presented at the same time or not. In [32], the authors propose HMBS (Hypermedia Model Based on Statecharts). The model makes use of the structure and execution semantics of statecharts to define the browsing semantics and the structural organization of multimedia documents.

The graph-based paradigm requires knowledge of those formal notations to specify the behavior of multimedia documents. Moreover, the task can become complex when authors need to define temporal relations among media segments (anchors), because each anchor must be defined as a different vertex in the graph. In addition, it makes user interaction specification harder than other paradigms such as the event-based one, as will be discussed in Section 2.1.6.

2.1.4 Hierarchy-based or Structure-based Paradigm. As far as the structure-based paradigm [16, 19, 34, 42, 59, 85] is concerned, it uses compositions to define temporal relationships among media items. There are different composition types: sequential, parallel, and atemporal. All items that are part of a sequential composition are presented sequentially. On the other hand, those that are part of a parallel composition are presented simultaneously. The atemporal composition [48] is defined as a grouping of components with no associated temporal relations among them, which will be presented when the composition is activated at runtime by a temporal composition. The SMIL *excl* element [9] is an example of atemporal composition. Using these compositions, authors can define the document synchronization as a tree of temporal containers.

In the structure-based paradigm, authors can also define a delay for media items and synchronization constraints among them. Such constraints can be defined in attributes of media assets or composition nodes, or in external structures, as CMIF's (CWI Multimedia Interchange Format) sync arcs [48, 85]. The structure-based CMIF model consists of nested presentations, events and channels. *Event* refers to a fragment of media data and a channel groups events of the same media type and their properties. The structure-based paradigm provides a simple synchronization model, in which authors can create a storyboard using the document structure. Also this paradigm allows authors to put the structure along a timeline as in CMIFed [85].

The concepts introduced by CMIF were applied to the SMIL language [24]. Thus, the authoring languages SMIL and MPEG-4 XMT [39] are also based on the hierarchical paradigm. The hypermedia model presented in [76] and the ZYX model [15] also use compositions to define temporal relations as a tree model representation.

Just as with the graph-based paradigm, relations among anchors are also hard to be represented in the structure-based approach, because a whole media item (and not only one of its anchors) must be included in a composition. For multimedia documents with several nodes and temporal relations, the document specification can become complex with several nested compositions. The hierarchical structure is also limited concerning representing synchronization conditions that are composed of more than one type of event or state. Events can refer to the presentation of media items, which can be in different states, such as occurring (media is playing), sleeping (media is stopped) or paused; and/or to user interactions. To overcome this limitation, SMIL State [51] extends the hierarchy-based synchronization providing state variable manipulation.

2.1.5 Constraint-based Paradigm. The constraint-based paradigm specifies temporal constraints among media items. These constraints are classified in two categories: reference point-based and interval-based synchronization. The latter defines the duration of an item as an interval. The interval synchronization is defined based on 13 relations between intervals discussed in [7]. Another approach can be based on 29 interval relations proposed in [86]. As in the hierarchical-based paradigm, the constraint-based synchronization is limited for defining user interaction and temporal relations among anchors. Madeus [54] and MPGS (Multimedia Presentation Generator System) [12] are authoring systems whose temporal synchronization model makes use of the interval-based paradigm. Regarding the point-based synchronization, it uses reference points to define the temporal synchronization among media items. These points can be the beginning and end of an item or anchor. As a result, reference point-based synchronization is more flexible than interval-based synchronization. Firefly [17] is an example of an authoring system that uses a

reference point-based model. The system has a hybrid synchronization model since it also uses the graph-based paradigm.

2.1.6 Event-based Paradigm. Regarding the event-based paradigm, it is based on event occurrences during multimedia application execution to define temporal relations among media items. Events can be classified in different types, such as presentation, selection, attribution and preparation events. The first one refers to starting, stopping and pausing a media item presentation. The selection event represents user interaction. The attribution event corresponds to changing variable values. The preparation event [53] allows buffering part of media content in the player in order to avoid delays when media presentation starts. The event-based paradigm is less intuitive for authors when compared to the timeline-based paradigm. As in the structure-based paradigm, multimedia documents with several media items and event-based relationships among them can make the document specification complex.

However, the event-based approach is very expressive for defining temporal relationships among media items [14]. This is very useful when multimedia applications are designed to be delivered on the Web since media items may suffer delays due to server and network load. This approach is also easily extended to new synchronization types [14], such as events with duration. The event-based paradigm also easily handles asynchronous events, such as user interactions and variable tests, which are not possible to be represented using the timeline-based paradigm. Hypermedia models that use the event-based paradigm are NCM (Nested Context Model) [79, 80], on which NCL (Nested Context Language) [72] is based, Labyrinth [35], and SIMM (Simple Interactive Multimedia Model) [30]. NEXT [67] and Composer [47] are multimedia authoring tools that are based on the NCM model. The STEVE editor [30] is based on the SIMM model. SMIL 2.0 [9] also uses event-based synchronization to specify a media object *begin* and *end* attributes based on a different media object begin or end time.

2.2 Authoring GUI Approaches

The study presented in [18] provides an authoring paradigm taxonomy that is not based on temporal synchronization models, as we discussed in Section 2.1. The authors instead base their approach on user interfaces for authoring multimedia documents. That is, they use the term *authoring paradigm* to refer to the GUI (graphical user interface) approach used by authoring tools to display the multimedia document structure to users.

In contrast, in this article we do not present multimedia authoring tools according to this classification due to the following reasons. First, this taxonomy does not take into account that graphical user interfaces (GUI) approaches (that is, authoring paradigms according to [18]) may not directly represent the underlying temporal synchronization paradigm of the conceptual multimedia model that supports tool implementations. The second reason lies in the fact that an authoring tool GUI may offer more than one authoring paradigm regardless of the underlying temporal synchronization paradigm. Third, our goal is to concentrate on features presented in multimedia authoring tools [18, 60] to support our discussion regarding mulsemedia authoring tools.

Therefore, we analyze authoring tools highlighting their temporal synchronization paradigm and authoring GUI approach separately. For the authoring tools of Section 2.3, this paper indicates the temporal synchronization paradigm according to Section 2.1. In respect of authoring GUI approaches, we classify them in textual, structural, layout, spatial, temporal view editing, wizard approach and/or form-based interface.

2.2.1 Textual View Editing. The textual view editing corresponds to directly writing multimedia documents using a standard multimedia language or script. The tools that follow this approach [16, 34, 40, 47, 54, 66, 67] provide a text editor integrated with other views. That is, the text editor

reflects its changes on the other views and vice versa. The textual view editing fits the model whereby authors with programming skills need to specify spatial and temporal behaviours that they cannot express using the graphical interface provided.

2.2.2 Structural View Editing. The structural approach refers to giving an editable graphical representation of media items and the relationships among them. This representation can be given using a tree, a set of compositions or graphs. The multimedia authoring tools SMIL Builder [16] and NEXT [67] use a tree representation for giving a structural view, but only SMIL Builder allows the editing of the tree. NEXT also provides an editable graph view that allows users to specify nodes and links. GRiNs [19] and SMILAuthor2 [90] use compositions (sequential and parallel) according to the discussion of Section 2.1.4. The remaining tools in Table 1 that provide structural view editing use graph representation.

2.2.3 Layout View Editing. Regarding the layout view editing, it allows users to edit the regions where media items are presented. It is important to highlight that this view only presents the beginning positions and dimensions of media items. However, those media properties can be changed during multimedia presentation, for example using NCL links [72]. The Composer [47] and NEXT [67] authoring tools provide the layout view editing of NCL documents. MediaTouch [59] also gives this editing ability allowing users to define presentation characteristics of MHEG-5 Scenes.

2.2.4 Spatial View Editing. In contrast to layout view editing, the spatial approach allows authors to visualize and specify the position of media items and how (e.g. size, style, volume etc) they are presented for each time instant including the beginning values of their presentation properties. To do this, authoring tools [30, 34, 42, 54, 80, 85] provide the spatial view integrated with the temporal view.

2.2.5 Temporal View Editing. Providing temporal view editing, authoring tools [30, 34, 54, 80] allow users to manipulate media items directly on the time axis. In other words, this view graphically presents the temporal order that media items are presented and their duration using a rectangle-based representation. In this view, authors can specify the beginning/end time instant of media items and their duration by manipulating these rectangles.

2.2.6 Wizard Approach. The wizard approach guides users through a sequence of steps that allows them to input information in an ordered way to produce a multimedia application as result. The NEXT editor [67] is the only one in Section 2.3 that offers this approach. It displays several windows in sequence guiding users to choose a multimedia application template and to select their media content to compose this template.

2.2.7 Form-based Interface. Similar to the wizard approach, the form-based interface allows users to input all the information for specifying the spatial and temporal behaviour by filling empty fields. However, the form-based approach does not guide users through a step-by-step process. From the tools discussed in Section 2.3, only MPGS [12] uses this GUI approach.

2.3 Authoring Tools

This section gives an overview of multimedia authoring tools available in scientific papers to identify a set of features for enhancing multimedia authoring. We also cite some commercial tools to support our analysis. In Section 2.4, we summarize all these features. Then the next section presents a table associating the authoring tools discussed with these features.

2.3.1 Academic Tools. STEVE [30] is a spatio-temporal editor for authoring interactive multimedia documents. It is based on its own event-based model called SIMM (Simple Interactive Multimedia Model). The model gives a causal interpretation to Allen's temporal relations [7]. STEVE thus provides a temporal view that allows authors to graphically synchronize media items using these temporal relations. In addition, STEVE allows users to specify media presentation properties and verify how and where media items will be displayed during document execution through a spatial view. The editor also provides the definition and simulation of user interaction events. Moreover, it gives authors feedback regarding temporal consistency. The editor can not only export multimedia applications to NCL documents but also to HTML5 files. In fact, STEVE translates NCL documents to HTML5 using NCL4WEB [78].

NEXT (NCL Editor supporting XTemplate) [67] is also a graphical editor for authoring NCL applications. It allows users with no knowledge of multimedia authoring languages to create NCL documents using templates. To this end, NEXT uses composite templates defined in XTemplate [36]. However, creating new templates is a hard task for non-programmers. As in Composer, users must also know NCM entities in order to use all features provided by NEXT.

Composer [47] is another NCL document authoring tool that provides different document views. However, the last version of Composer does not offer a temporal view. Since the NCM model directly underlines the Composer's graphical environment, users must know NCM entities, which is a non-trivial task. Authors should use the structural view to specify their documents by manipulating icons and rectangles that represent NCM nodes and links.

HyperProp [80] is a hypermedia system that consists of an authoring tool and a formatter. The event-based NCM model [79] underlies the system. Therefore, the HyperProp authoring tool provides a structural view for specifying links among nodes based on events. Icons (content nodes) or rectangles (composite nodes) and lines represent NCM nodes and links, respectively. The tool also offers a temporal view integrated with the structural one. Moreover, it provides a spatial view allowing authors to define spatial relationships among nodes and their presentation properties. Authors can also receive feedback regarding temporal and spatial consistency.

CMIFed [85] provides three document views that give graphical representations of the CMIF's elements. The hierarchy/structural view represents the nested presentations and events using nested sets of boxes to express temporal compositions. The channel view (temporal view) offers a visualization of time flows from top to bottom separated by channels. However, the editor does not allow users to edit this temporal view. CMIFed also provides a player that allows the editing of layout aspects of presentations. Users can also preview a presentation or part of it directly from the hierarchy and channel views. Moreover, the editor allows users to debug presentations by playing and checking when events are active in the channel view.

GRiNs [19] is an extension of CMIFed by implementing navigation facilities. It consists of an authoring and presentation system for SMIL [24] documents. The system uses a temporal synchronization model based on parallel and sequential compositions. Its structural view shows nested rectangles to represent SMIL compositions that give authors the temporal synchronization. However, this representation may become difficult to be understood when applications have several temporal compositions nested on many levels. The tool also provides a timeline view but only for visualizing.

SMIL Builder [16] is also an editor for creating SMIL documents with incremental verification based on a hierarchical SMIL Petri Net model. The tool offers textual and structural views to edit SMIL documents. The structural view allows the specification of SMIL documents using a tree structure. The textual and temporal views display any modification in the structural one. Regarding the temporal view, it does not allow editing and is only for visualization. Moreover, it does not

display the media items directly on the time axis. Instead, it uses graphical elements to represent the Petri-net-based model that underlies SMIL Builder.

SMILAuthor 2.0 [90] is another tool for exporting SMIL documents. It provides a structural view using compositions as well and supports non-deterministic temporal behaviour. Moreover, SMILAuthor 2.0 provides a layout view to edit spatial relationships among media items. The editor also offers a limited presentation preview, in which users can only preview part of the presentation by specifying the preview duration.

LimSee2 [34] is a SMIL authoring tool that provides both spatial and temporal views. The spatial view allows users to edit SMIL regions by moving and resizing media items. In the temporal view, authors can synchronize media items by dragging and resizing them to define their start/end time and duration. However, the tool does not support user interaction in the document definition. Besides, *LimSee2* requires a basic knowledge of the hierarchical model to synchronize media items temporally.

FireFly [17] is an authoring tool which uses a constraint-based model and presents a structural view based on graphs. It also uses events to synchronize media items. However, this structural view may make the document temporal behavior verification difficult since media items are not directly placed in a time axis. This tool neither provides spatial view editing nor does it generate an application using a standardized format.

Madeus [54] is a multimedia authoring tool that allows spatial and temporal editing using a constraint-based specification. The tool provides a graphical temporal view in which users can check not only the temporal order of media items but also the constraint relations among them. Authors can also edit this view moving or resizing a media item along the horizontal axis. However, the editor requires users to create a multimedia document using the Madeus textual format before editing it through the graphical views. In the presentation view, authors can play a document, pause it and edit the spatial position of media items.

MPGS [12] is a multimedia presentation system that is underpinned by a constraint model. The system consists of two environments for specifying and generating the presentation. The specification environment allows users to define temporal and spatial constraints between two objects by using a form-based approach. The system also performs spatial and temporal consistency checks during the authoring phase and afterwards users complete the specification. Moreover, MPGS uses strategies to generate presentations when spatial and/or temporal constraints specified by users cannot be satisfied at run-time.

MediaTouch [59] is an authoring tool for creating MHEG-5 [37] objects. It provides structural and layout view editing. MediaTouch uses a structure-based paradigm for synchronizing media items and offers interactivity support through the Link Editor interface. The tool also provides a player.

Gaggi and Celentano [42] propose a tool that allows users to create multimedia presentations with parallel and sequential synchronization, and hyperlinks for navigation. It offers structural view editing by using a tree representation and a graph view. It also provides a spatial layout view and a preview. However, neither a standard language is used to export applications nor a player is provided.

Videobook [66] is a prototype hypermedia system that uses a script-based model to specify the presentation layout and display timing. The system is composed of a database, editors and a player. The database stores the scripts associated with each model component, such as media, triggers (buttons) and scene (set of media and triggers). The system provides the scene Editor, which is the center of authoring. This editor provides a graphical representation of media alongside triggers over time and space according to their scripts. Videobook thus gives only a presentation visualization and does not allow users to graphically specify the multimedia presentation. To specify the scripts,

Videobook provides a text editor. The system also offers a player that creates a schedule table using the scene scripts.

Harmony [40] is a multimedia presentation system that provides a database to store multimedia content and an authoring environment, the Harmony user interface. The interface allows user to create media nodes and links among them through a script-based interface. It also supports interactivity events to trigger links and displays the structure of multimedia documents in a tree graph, which illustrates the temporal synchronization among the media. However, this tree cannot be edited in order to specify the behaviour of multimedia applications.

I-HTSPN [89] provides a graphical interface for creating multimedia documents based on Timed Petri nets. To build nets, users can graphically create places, transitions and arcs. Authors can also specify attributes associated with these elements using dialog windows. In addition, a player is provided to run MHEG multimedia applications. The editor has also been implemented to produce Java multimedia applications.

Trellis [41] provides a graphical editing client and two other clients for displaying text and graphics images content in two independent windows. The editor represents multimedia applications using a Timed Petri-net-based view. With this view, users can edit the structure of the net through graphical elements that illustrate each component of the net, such as place, transition and token.

caT [65] extends Trellis [41] in order to support context-aware documents that respond to environment changes, such as time, location and bandwith/cost. To do that, caT provides user modeling, high-level Petri-net specification and fuzzy knowledge. The fuzzy logic engine is invoked to infer right values from uncertain user contexts. The caT editor provides multiple subnets status in the simulation node to aid users check the document presentation. The caT system also features an analysis tool that helps authors to verify the document behavior and even offers an interactive debugging tool.

2.3.2 Commercial Tools. There are also commercial software solutions that allow users to create video presentations, such as Final Cut Pro [8], Adobe Premiere [4], Davinci Resolve [13], Adobe Director [3], and Nero Video 2021 [5]. They use the timeline-based paradigm as discussed in Section 2.1 and are designed for video editing professionals. Thus, they offer a multitude of features and menus that make the creation process hard for ordinary users. These commercial tools do not produce hypermedia applications written in a standard multimedia authoring language either. Usually, they encode multimedia applications in video file formats, such as MP4 and MOV.

2.4 Desirable Features for Multimedia Authoring Tools

From the discussion regarding both academic and commercial authoring tools, a set of features can be identified which need to be provided by multimedia authoring tools in order to enhance the multimedia production phase. They are as follows:

- *Temporal View Editing*: tools need to provide a temporal view for presenting the temporal behaviour of media items. The items have to be directly placed on the time axis so that users can verify when media items are presented and their duration. Additionally, tools must allow users to edit the temporal view by changing the duration and beginning/end time instant of media items.

- *Spatial View Editing*: a spatial view must be provided by tools for presenting how and where media items are visually presented during execution. This view needs to be synchronized with the temporal view so that users can check the spatial view for each time instant of the multimedia presentation. Tools also need to provide user interface commands to allow

authors to graphically define media properties such as size, position etc.

- *Interactivity Support*: authors need to be able to define interactivity relations. Those relations refer to user interactions using input devices (e.g. keyboard, mouse, remote control, microphone, camera, touch screen, eye tracker, etc) with a media item. In other words, authors must be able to define an interactive media, with which users can interact, and actions must be triggered from that user interaction.

- *Presentation Preview*: authors need to visualize the temporal and spatial behaviour of their multimedia applications during the authoring phase before publishing them. This visualization should give a graphical representation of the application layout over time according to the temporal synchronization. In case of interactive multimedia applications, the preview may not provide user interaction. In other words, the preview is a simulation and not the application execution itself.

- *Ordinary User Support*: tools should allow authors with no knowledge of multimedia language and/or model to create interactive multimedia applications. To put it differently, tools should offer a graphical user interface approach that can guide authors in their production phase.

- *Template Support*: templates can be defined as generic structures that specify the spatio-temporal behaviour of multimedia documents but without defining media content. Tools should provide templates so that users just have to complete a template definition with media content in order to produce whole multimedia applications.

- *Standard Publishing Format*: authors should be able to export their multimedia applications to documents using international standard multimedia languages, such as HTML5 and NCL (SMIL W3C Group closed its activities in 2012 [44]). This feature is more flexible than exporting an application to a video file, where user interaction is not possible.

- *Content Delivery Analysis*: the analysis of performance regarding the delivery of multimedia content is essential in networked multimedia systems. Even in the authoring phase, tools should provide authors about feedback related to the delivery performance of media items added in their applications. Additionally, tools may also provide optimization methods for different types of media content delivery.

- *Error Analysis*: authors should have feedback about inconsistent spatial or temporal behaviours when they create or edit applications. This feature is very useful in order to avoid execution errors [75] after applications are distributed to final users. Authoring tools based on formal models, such as the ones based on the graph-based paradigm discussed in Section 2.1.3, have advantages considering this feature.

- *Player*: tools need to provide an embedded multimedia player or to be integrated with an external player so that authors can fully interact with their multimedia applications in order to experience what users will in a near future.

2.5 Comparison of Multimedia Authoring Tools

Table 1 provides a summary for multimedia authoring tools discussed in Section 2.3.1 regarding the features presented in Section 2.4. In addition to those features, the table also indicates the

temporal synchronization paradigm of each tool according to the paradigms discussed in Section 2.1. Concerning the temporal and spatial view editing feature, they are represented in the column *Authoring GUI Approach* using the words *temporal* and *spatial*. That column indicates which graphical user interface approaches the tools provide to allow authors to graphically create multimedia applications. The authoring approach types are in keeping with the discussion of Section 2.2.

These multimedia authoring tools that we discussed cater for traditional audiovisual-based multimedia. However, olfactory, wind and thermal effects in mulsemedia applications are characterised by a new set of characteristics such as wafting and lingering, which affect intensity [6, 43, 62] in contrast to traditional media (intensity of audio volume or image brightness). Therefore, new requirements raised by mulsemedia applications are insufficiently catered by legacy multimedia editors. The next sections discuss several studies that focus on the mulsemedia authoring phase in order to support us to identify this new set of features to be provided by mulsemedia authoring tools.

3 MULSEMEDIA MODELING

In the next subsections, we highlight studies that support the representation of sensory effects and their characteristics. Those proposals involve conceptual mulsemedia models, programming frameworks, description and programming languages. Besides contributing to the mulsemedia production phase, they also support the development of mulsemedia authoring tools.

3.1 MPEG-V

The MPEG-V standard [56] defines a set of XML-based elements for specifying real-world objects, such as sensors and actuators, and virtual world objects. These elements aim at standardizing the data exchange between both worlds. MPEG-V defines *Control Information Description Language* (CIDL) to represent metadata related to actuator and sensor capability, user's sensory effect preference and sensor adaptation preference. The standard also provides the Sensory Effect Description Language (SEDL) for describing sensory effects.

An SEDL file, which is called Sensory Effect Metadata (SEM), annotates multimedia content with sensory effects. To define the temporal synchronization between sensory effects and audiovisual contents, MPEG-V uses a timeline-based paradigm in which each effect has its beginning and ending time synchronized with a specific audiovisual content. This temporal model has several well-known limitations [14], mainly concerning the user interaction, as discussed in Section 2.1.2.

The standard also allows the specification of sensory effect intensity fade-in and fade-out in time, which respectively defines when effects reach their maximum intensity and null value. The intensity specification of sensory effects is essential for authoring mulsemedia applications since it allows authors to define how strong sensory effects are rendered in real environments [52]. Additionally, the specification of fade-in and fade-out for sensory effects allows authors to define an intensity variation over time increasing the quality of experience (QoE) of mulsemedia applications.

With regard to the spatial model of MPEG-V, sensory effects have the attribute *location*. This attribute specifies the region where effects should be perceived by users that are immersed in the mulsemedia content. The spatial model considers the user as the central point of reference and the effect location is defined using x, y and z axes.

Furthermore, the SEDL language provides elements for grouping effects so that authors can create complex effects, for instance an explosion that may include wind, light and vibration effects. The language also allows authors to reuse sensory effects that are frequently used by declaring and referring them. Those sensory effects (SEM files) should be transformed into actuator commands in the real world to render them using a virtual world to real world (VR) adapter, which is out of the scope of MPEG-V.

Table 1. Features of Multimedia Authoring Tools

Feature / Tool	Temporal Synchro-nization Paradigm	Authoring GUI Approach	Interactivity Support	Presentation Preview	Ordinary User Support	Template Support	Publishing Formats	Content Delivery Analysis	Error Analysis	Player
STEVE [30]	event-based	temporal spatial	✓	✓	✓		NCL HTML5		✓	✓
NEXT [67]	event-based	layout structural textual wizard	✓		✓	✓	NCL			
Composer [47]	event-based	layout structural textual	✓				NCL			✓
HyperProp [80]	event-based	structural temporal spatial	✓				NCM HTML		✓	✓
CMIFed [85]	structure-based	spatial structural	✓	✓			CMIF			✓
GRiNs [19]	structure-based	structural	✓	✓			SMIL			✓
SMIL Builder [16]	petri-net-based	structural textual					SMIL		✓	
SMILAuthor2 [90]	petri-net-based	structural layout	✓	+/-			SMIL			
Limsee2 [34]	structure-based	temporal spatial textual		✓			SMIL			✓
FireFly [17]	event constraint -based	structural	✓				own format			✓
Madeus [54]	constraint-based	textual temporal spatial	✓	✓			Madeus SMIL		✓	✓
MPGS [12]	constraint-based	form-based					own format	✓	✓	✓
MediaTouch [59]	structure-based	structural layout	✓				MHEG-5		✓	✓
Gaggi [42]	structure-based	structural spatial	✓	✓			own format		✓	✓
Videobook [66]	script-based	textual	✓	✓			own format			✓
Harmony [40]	script-based	textual	✓				own format			✓
I-HTSPN [89]	graph-based	structural	✓				MHEG Java		✓	✓
Trellis [41]	graph-based	structural	✓				own format		✓	✓
caT [65]	graph-based	structural	✓	✓			own fmt. HTML		✓	✓

✓ : fully supported feature; +/- : partially supported feature

In addition, MPEG-V allows the description of virtual objects, which are classified in general objects and avatars. For both types, the standard defines properties, such as identity, audio resource, scent, movement controls and input events, to allow those objects to be controlled by the real world

inputs and to be used by different virtual worlds. Note that MPEG-V only defines virtual object properties and does not define their geometric shape, animation and texture. To support these last specifications, MPEG-4 [68] may be integrated with MPEG-V.

With IIDL (*Interaction Interface Description Language*), MPEG-V defines two vocabularies: *Device Command Vocabulary* (DCV) and *Sensed Information Vocabulary* (SIV). DCV allows the description of actuator commands to render sensory effects in the real world. On the other hand, SIV is responsible for modelling information captured by sensors.

Although it provides a set of XML-based languages, MPEG-V does not support the definition of an entire mulsemedia application as can be done with MultiSEM [31] and Guedes' framework [46]. In other words, MPEG-V cannot represent multiple types of media, such as video, image and text, and synchronize each of them with several sensory effects in time and space. Indeed, MPEG-V only specifies temporal annotations of sensory effects for a single video or audio.

3.2 MultiSEM

MultiSEM [31] (Multimedia Sensory Effect Model) is an event-based mulsemedia model for integrating and synchronizing multiple sensory effects with traditional multimedia content in interactive mulsemedia applications. The model represents sensory effects as document nodes following the approach presented in [52], which models sensory effects as first-class entities. This representation approach uses a high-level abstraction so that the spatio-temporal synchronization of mulsemedia applications can be specified regardless of devices used for implementing mulsemedia applications in the real world.

The event-based paradigm used by MultiSEM is applied to several proposals of multimedia authoring tools [30, 47, 67] and models [35, 79, 80] since the paradigm is more expressive [14] as we discussed in Section 2.1. MultiSEM also uses the concept of hypermedia connector [64] to represent spatio-temporal relations among nodes, whether traditional multimedia content or sensory effects. The model thus allows effects to be temporally synchronized with other nodes, media or sensory effect nodes, according to the occurrence of events in mulsemedia applications.

Furthermore, MultiSEM is based on *Part 3: Sensory Information* of MPEG-V [84] to define the sensory effect entity and their subclasses (e.g. wind effect, heat effect etc). The model also provides properties to represent rendering characteristics of effects, such as intensity value and range (e.g. *lux* unit is used for light effects), position and specific rendering properties of each effect type. The MultiSEM spatial model for sensory effects is also based on the MPEG-V standard. Effects can therefore be located in physical environments according to the X, Y and Z axes [56].

3.3 Guedes et al.

Guedes et al. [46] propose a high-level programming framework in order to support multimodal user interfaces for multimedia applications. The authors propose the integration of concepts from the multimedia and Multimodal User Interfaces (MUIs) communities. The framework supports different types of input and output modalities.

Regarding input modalities, the framework provides user-generated input modalities, such as gestures and voice recognizers. For instance, it uses SRGS (*Speech Recognition Grammar Specification*) [50] files to define which speech should be recognized. In other words, these files define recognizer anchors to specify parts of the recognizer content. With respect to output modalities, it offers traditional audiovisual content, speech synthesizers and actuators. Those output modalities can stimulate different human senses (hearing, smell, touch, taste, or vision) using audiovisual devices and actuators to render different kinds of sensory effects.

To allow users to synchronize recognizers and synthesizers, the framework defines four actions. *Start*, which enables the recognizer or synthesizer; *stop* to deactivate them; *pause* to deactivate without releasing its resources; and *resume*, which reactivates a paused recognizer or synthesizer.

Additionally, the framework defines a user class base aiming at identifying user profiles and allowing applications to adapt according to these profiles. In other words, the authors propose contextual elements to provide the modality selection based on the user sensory capabilities (see, speak and hear). To complement this adaptation, the framework also considers environment characteristics with the user description to make the modality selection. Furthermore, the authors present a case study using the NCL language to illustrate the proposed framework.

The main difference between the framework presented in [46] and MultiSEM [31] is the abstraction level used for representing sensory effects in mulsemedia applications. In [46], the authors provide sensory effects by representing sensors and actuators as nodes. On the other hand, MultiSEM models sensory effects as document nodes in a higher-level abstraction.

3.4 ASAMPL

ASAMPL (Algebraic System of Aggregates and Mulsemedia data Processing Language) [69, 82, 83] is a programming language for enabling the authoring of mulsemedia applications with multimodal content. The language focuses on representing complex information composed of multiple data of different types. The main idea is to represent a real scene using not only audiovisual data but also olfactory and taste data. ASAMPL also represents physical data of objects or environments, such as air, water, metal, and their properties, temperature, pressure, density to describe the real world.

The language defines the concept of multi-image to represent multimodal information by applying the Algebraic System of Aggregates' concepts. A multi-image consists of a set of muxels (multimodal elements). The muxel term is defined based on voxel graphics, a volume element representing 3D objects. A tuple of values describes a muxel representing different modality of the object characteristic in this point (muxel) in a specific instant. However, authors must use external sources to give the details of the multimodal content. Indeed, they must use the ASAMPL's Source and Download statements to assign these details to a tuple or aggregate (set of tuples).

ASAMPL also provides built-in commands that allow developers to fulfill necessary actions on multimodal data. Using these actions, developers can download and upload data streams, temporally synchronize data of different modalities using a timeline-based paradigm and change the data duration.

Figure 1 illustrates four quadrants defined according to temporal synchronization paradigm (event-based and timeline-based) and abstraction level to give an overview of the mulsemedia models discussed in this section. The horizontal axis ranks the models according to their abstraction level for representing sensory effects and traditional multimedia. Models that are closer to the right have a higher abstraction level than those closer to the left. For example, MultiSEM has the highest abstraction level since it is most to the right. On the opposite side, MPEG-V standard has the lowest abstraction level for representing sensory effects.

The vertical axis classifies those models into two groups. One of them defines the event-based models. This group contains the proposal of Guedes et al. [46] and MultiSEM since the models use the event-based paradigm to temporally synchronize mulsemedia contents. The other group contains timeline-based models, MPEG-V and ASAMPL. For instance, ASAMPL is a timeline-based model classified as a low-level abstraction model. However, this model has an abstraction level higher than MPEG-V.

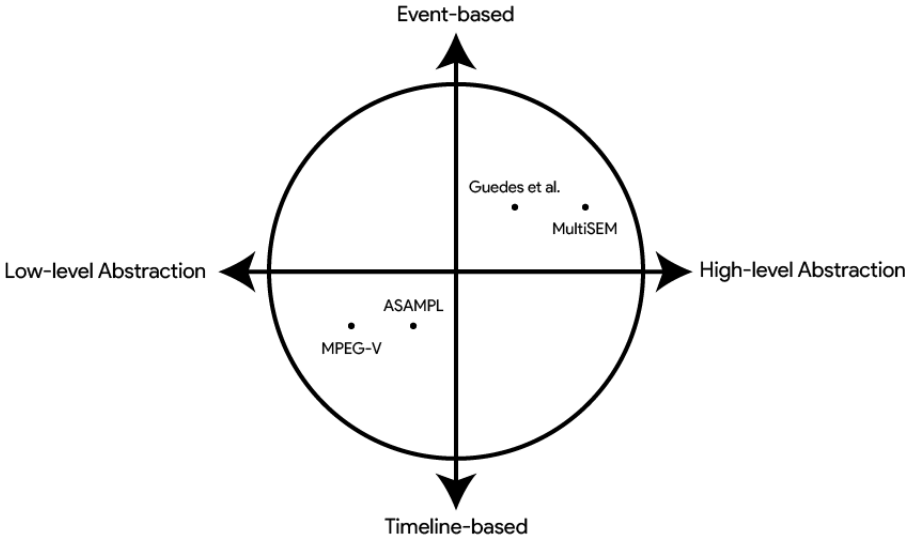

Fig. 1. Mulsemedia Models Overview

4 MULSEMEDIA AUTHORING TOOLS

In addition to the previous studies that help us represent sensory effects in mulsemedia applications proposing conceptual models or declarative languages, others focus on enhancing the mulsemedia authoring phase providing graphical tools. Those tools support the definition of sensory effect metadata and/or the specification of complete mulsemedia applications using different authoring GUI approaches. Here we present different mulsemedia authoring tools, discussing their features and comparing them so that we can raise requirements and challenges regarding graphical authoring tools.

4.1 STEVE 2.0

STEVE 2.0 [31] is an extension of STEVE [30], discussed in Section 2.3.1, to support the integration and synchronization of sensory effects with traditional media. It uses the MultiSEM model, discussed in Section 3.2, as its underlying model to represent mulsemedia applications and to synchronize their nodes, whether traditional media or sensory effects. Therefore, the tool uses an event-based temporal synchronization paradigm. STEVE 2.0 graphically provides causal, temporal relations based on MultiSEM's relations to allow users to synchronize mulsemedia documents. It also gives authors feedback about temporal synchronization inconsistencies. Users can also create interactivity relations with STEVE 2.0 since it is based on MultiSEM. These relations allow authors to activate, for example, a sensory effect due to user interaction with the mulsemedia application. The editor also provides spatial view editing to allow users to edit rendering characteristics and physical positions of sensory effects. Users can also edit presentation properties of traditional multimedia. Since the tool provides a temporal view as the authoring GUI approach, users do not need to have programming skills to define a mulsemedia application using STEVE 2.0 environment. Moreover, STEVE 2.0 provides automatic extraction of sensory effects from audiovisual content analyses by applying Machine Learning-based methods [1]. However, STEVE 2.0 does not provide final formats that can be run in mulsemedia players yet, since the tool implementation is a work in progress.

The tool also does not offer a simulator, although it allows users to check the temporal behavior of sensory effects together with traditional multimedia.

4.2 CrowdMuse

Aiming at finding out whether a crowdsourcing approach can support mulsemedia authoring, particularly sensory effect annotation associated with a specific video, authors in [29] present a web platform, called CrowdMuse. This platform collects sensory effect annotations from the crowd allowing authors and users to work together in the mulsemedia authoring phase. However, CrowdMuse does not focus on providing a fully authoring environment. Therefore, the tool does not allow the synchronization of multiple media and sensory effects, rendering characteristics editing and does not provide a simulator or player. Instead, CrowdMuse focuses on collecting the contributions of users regarding video time intervals that they think to be appropriate to associate sensory effects with. As the outcome of the crowdsourcing process, the tool exports the sensory effect annotation to the MPEG-V format so that other players compatible with this format can reproduce those effects.

4.3 H-Studio

H-Studio [28] is an authoring tool that focuses on the creation of only two types of sensory effects: vibration and motion. The tool is composed of three main parts: a video preview to play the video to be annotated with sensory effects, a timeline where users can synchronize sensory effects with the video and a menu to allow authors to define parameters of the current sensory effect selected in the timeline. For vibration parameters, H-Studio presents only two parameters: amplitude and frequency. On the other hand, the tool provides a more complex interface for defining the properties of motion effects according to the Six Degrees of Freedom (6DoF) of a rigid body in three-dimensional space. It means that an object can move forward/backward, up/down, left/right combined with rotation in three perpendicular axes. H-Studio offers three methods to support the specification of 6DoF movements. The first one allows authors to use a force-feedback device, the second defines the motion through a trajectory recording from a force-feedback device and the last method allows users to import those parameters from the real world. To preview the effects created, the tool provides a video player that can be synchronized with a force-feedback device that renders the effects.

4.4 Sang-Kyun Kim et al.

In [58], authors propose a method to extract temperature sensory effects from audiovisual content. To this end, the method consists of extracting the colour temperature of scenes and mapping their properties to the temperature effect attributes. They also introduce an authoring tool to apply the proposed method. The tool contains three parts: a video control component, a temperature effect component and an effect sensory timeline component. The video control component allows authors to go forward or backward frame by frame in a specific video. With the temperature effect module, authors can select frame intervals to extract the colour temperature. Then from colour temperature properties, the tool creates temperature effect metadata specified in MPEG-V. Moreover, the authoring tool provides a timeline for showing the calculated colour temperature categories and their correlated temperature effects along the time axis. Authors can also define frame intervals using this timeline.

4.5 SEVino

SEVino (Sensory Effect Video Annotation) [87] is a graphical tool that provides a timeline divided in channels. Each channel represents a type of sensory effect, such as wind, vibration, light etc.

In the timeline, users can create rectangles whose sizes define sensory effect duration. Also, the tool provides video frames along the time axis in order to allow authors to synchronize effects with a single video content. In addition, SEVino allows users to define some properties of the corresponding sensory effect selected on the timeline. Moreover, SEVino can not only export sensory effect annotation to MPEG-V but the tool can also import MPEG-V SEM description files so that users can modify or extend them using SEVino's GUI. In addition to these features, SEVino is integrated with a player, Sensory Effect Media Player (SEMP), and a simulator, Sensory Effect Simulator (SESim). Both of them are also proposed in [87] and will be discussed in more detail in Section 7.

4.6 RoSE

RoSE (Representation of Sensory Effects) Studio [21] is an authoring tool that also uses a timeline in order to synchronize sensory effects with audiovisual contents. As SEVino, RoSe exports sensory effects to SEM files using MPEG-V. In addition, RoSe multiplexes the effects using the MPEG-2 TS standard. RoSe's GUI is composed of a video player, a region for editing sensory effect properties and a timeline. The tool also offers a specific graphical interface to define the sensory effect position in the 3D space based on the MPEG-V spatial model. Although the tool provides several types of sensory effects, it does not allow users to synchronize them with several multimedia items. Instead, RoSe only supporting the sensory effect annotation with a single video.

4.7 SMURF

Another tool for creating MPEG-V SEM files using a timeline is SMURF (Sensible Media Authoring Factory) [55]. It is also based on the MPEG-V standard and its GUI is composed of a video player, a region for defining the sensory effect rendering characteristics and a timeline for synchronizing sensory effect with a single video. Additionally, SMURF provides an interface to create groups of effects. In this interface, users can define several types of effects to compound a complex effect, for example, an explosion effect that can involve temperature, light, flash and wind effects. As a result, only one effect represents the explosion effect, rendering different types of basic effects. Furthermore, SMURF allows authors to reuse a declared frequently-used effect. Last but not least, the tool can import and export MPEG-V SEM files.

4.8 Real 4D Studio

In [77], authors propose an authoring environment composed of three tools: Real4DAExtractor, Real4DEmaker and Real4DAStudio. The first is responsible for extracting rigid body motion, light and flash effects from visual contents through frame analysis. Real4DEmaker allows users to generate SEM files for each sensory effect at a time. Users can also import a SEM file and simulate the effect in a 3D virtual world. In addition, Real4DEmaker provides an interface for editing effect position based on the MPEG-V spatial model. Regarding Real4DStudio, it allows authors to synchronize sensory effects created in Real4DEmaker with a single video using a timeline approach. A 3D simulation for checking the synchronization of several effects is also available in Real4DStudio. Additionally, this tool presents two types of interfaces according to the purpose of usage: media-based and event-based interface. Both of them use a timeline approach. However, the media-based interface provides authoring targeting specific media. The event-based interface provides an effect preview instead of a canvas view to focus on the authoring of a particular sensory effect.

4.9 MulSeMaker

MulSeMaker [33] aims at enhancing the authoring of mulsemedia applications for authors with no programming knowledge by providing a template-based interface approach. It uses web components to define custom XML tags for defining and integrating sensory effects with HTML media objects. This integration uses a timeline-based paradigm for synchronizing continuous media with sensory effects. For discrete media, it uses the event-based paradigm. In this case, users can define interactivity events, such as a user click to start or stop sensory effects. Also, users can start sensory effects when a specific discrete media begins its presentation. MulSeMaker's graphical interface is based on wizard and templates. Authors should select a template, choose media objects to be part of the document and define application sensory effects. For each effect, authors associate it with a media object and define its beginning and ending time by using a table. The use of wizard and templates support authors with no knowledge of programming. However, this approach limits the expressiveness of the authoring tool by restricting authors to the predefined template set. Moreover, authors are not able to modify the behavior of these applications. Therefore, Mulsemaker does not allow authors to define new mulsemedia applications by defining document nodes and spatio-temporal relationships among them.

4.10 FeelReal

FeelReal [38] is a commercial tool for synchronizing sensory effects with a single video. The editor focuses on scent effects since its vendor provides a scent generator mask to integrate with several brands of virtual reality headsets. This mask can also render water mist, wind, heat and vibration effects. FeelReal provides an interface based on timeline and panel with different kind of scents that the mask can render. The editor also allows users to define the effect position but limited to the left and right sides, which corresponds to the sides of the mask device. It does not provide a simulator nor does it export the sensory effect description to a standard format, such as MPEG-V, due to the description being directly sent to the mask device.

5 DESIRABLE FEATURES FOR MULSEMEDIA AUTHORING TOOLS

Taking the discussion of mulsemedia authoring tools into account and based on multimedia authoring features presented in Section 2.4, we also identify a set of features to be provided by mulsemedia authoring tools for enhancing the mulsemedia production phase. They are as follows:

- *Temporal View Editing*: tools need to provide a temporal view for presenting the temporal behaviour of traditional multimedia items and sensory effects. Not only media items but also effects have to be directly placed in time axis so that users can verify when they are presented and their duration. Additionally, tools must allow users to edit the temporal view by changing the duration and beginning/end time instant of media items and sensory effects.

- *Synchronization of Multiple Media and SE (Sensory Effects)*: tools should allow users to temporally synchronize not only a single video but also several types of media, such as video, image, text and/or audio, with various sensory effects at the same time through a temporal view.

- *SE Group*: an interface for defining groups of effects so that authors can create complex effects using basic sensory effects should be provided. For instance, an explosion effect can be composed of the basic effects, such as wind, light and vibration effects.

- *SE Reuse*: tools need to provide the reuse of sensory effects that are frequently used. Authors should be able to declare and refer them.

- *Spatial View Editing*: a spatial view must be provided by tools for presenting and editing how (presentation and rendering properties) and where (media and sensory effect position) media items and sensory effects are presented or rendered during execution. It is noteworthy that editing rendering properties include changing the intensity value of sensory effects. As in the multimedia context, this view has to be synchronized with the temporal view so that authors can verify the spatial view for each time instant of the mulsemedia presentation.

- *Interactivity Support*: as in multimedia applications, authors also need to be able to define interactivity relations in mulsemedia applications to allow users to interact through an input device (e.g. remote control) with (generally visual) media items. In order to specify that feature, authors must define an interactive media or effect, with which users can interact, to trigger an action that can modify the temporal or spatial behaviour not only of other media items but also of sensory effects.

- *Statement Assessment Support*: this requirement refers to allowing authors to define statement assessment using information captured from sensor devices to trigger actions on media items or sensory effect(s) based on the particular sensed information.

- *Simulator*: tools should allow authors to visualize the temporal and spatial behaviour in an integrated simulation virtual environment so that authors can verify when, how and where media items and sensory effects would be presented in the real world before publishing the mulsemedia application. In case of interactive applications, the simulation may not provide user interaction.

- *Ordinary User Support*: tools should allow authors with no knowledge of mulsemedia language and/or model to create mulsemedia applications offering a graphical user interface approach that can guide authors in their production process.

- *Template Support*: templates are generic structures that specify the spatio-temporal behaviour of mulsemedia applications but with no definition of media contents and sensory effect annotation. In other words, tools should provide templates so that users just have to define media contents and sensory effect annotation for each media item that requires it in the template specification to create mulsemedia applications.

- *Automatic extraction*: tools should allow authors to automatically extract sensory effects from audiovisual contents to enhance sensory effect annotation.

- *Standard Publishing Format*: authors should be able to export their mulsemedia applications from the tool in which they were created to documents written in standard mulsemedia languages.

- *SE Prefetching Support*: for networked mulsemedia systems, the performance analysis concerning the sensory effect rendering in the real world is essential since delays can occur during the preparation of actuator devices [53], which is responsible for rendering sensory effects in the real world. Therefore, techniques [53] can be applied to configure the preparation of

rendering devices in the authoring phase to avoid delays in effect rendering.

- *Error Analysis*: authors should have feedback about inconsistent spatial or temporal behaviors when they create or edit mulsemedia applications. This feature is very useful in order to avoid execution errors [75] after applications are distributed to final users.

- *Player*: tools need to provide an embedded mulsemedia player or to be integrated with an external player so that authors can fully be immersed in mulsemedia applications and feel sensory effects in real life as final users will.

6 COMPARISON OF MULSEMEDIA AUTHORING TOOLS

Table 2 presents a comparison among the mulsemedia authoring tools discussed in Section 4. We compare them based on the functional requirements for mulsemedia authoring tools presented in the previous section. Additionally, the comparison table presents the temporal synchronization paradigm for each mulsemedia tool. The tools cited in this survey use the event-based and/or timeline-based paradigms, which follow the definition presented in Section 2.1. Furthermore the table describes the authoring GUI approach, which also follows the same classification presented in Section 2.2 although the mulsemedia tools discussed in this study only use the temporal, spatial or wizard approach.

In order to present a more detailed comparison regarding the spatial view editing, we divided this feature into four subtopics: *SE Position Editing*, *Rendering Editing*, *SE Spatial View* and *Multimedia Spatial View Editing*. The *SE Position Editing* feature allows authors to edit the 3D position from where sensory effects are rendered. Some of the tools discussed in Section 4 provide this feature based on the MPEG-V spatial model. H-Studio partially supports this feature since the tool only allows authors to edit motion effect position. FeelReal also partially provides position editing since authors can only define the left and right side of the headset to render effects. The *Rendering Editing* feature refers to an interface to allow authors to edit the several rendering properties that sensory effects may be characterized by. Those properties can also be based on the characteristics defined in MPEG-V. MulSeMaker and FeelReal partially provide this feature since both tools only allow the effect intensity editing. The tool presented in [58] also supports the rendering editing partially since it only allows authors to edit properties of temperature sensory effects. The *SE Spatial View* subtopic corresponds to the tool capability of allowing users to verify the spatial behaviour of sensory effects together with traditional multimedia content for each time instant synchronized with the temporal view. Finally, *Multimedia Spatial View Editing* indicates whether tools specifically provide a spatial view for traditional multimedia items, which is the case of STEVE 2.0.

Moreover, Table 2 indicates the *Asynchronous Event Support* feature, which includes the requirements related to the interactivity and statement assessment support described in Section 5. Those features correspond to events that may occur during the application execution and we cannot predict when they will occur during authoring time. MulSeMaker partially supports the interactivity feature. Although the predefined templates available in MulSeMaker may contain interactivity events already specified, users cannot create new interactivity relations or edit the existing ones.

The tool introduced in [58] partially provides the *Automatic Extraction* feature since it extracts only temperature sensory effects from visual contents. On the other hand, STEVE 2.0 and Real 4D Studio extract different types of sensory effects. Concerning the *Player* column, it indicates which tool provides a mulsemedia player. Each player will be discussed in Section 7. The last line of the table, *Feature Score*, provides a rank among the tools we discussed by checking how many features they support. For each feature the tools provide, they earn 1 point. For features that are partially supported, the tools earn 0.5 points. For example, H-Studio totally provides *Ordinary User Support*

Table 2. Features of Mulsemedia Authoring Tools

		STEVE 2.0 [31]	CrowdMuse [29]	H-Studio [28]	Kim et al. [58]	SEVino [87]	RoSE [21]	SMURF [55]	Real 4D Studio [77]	MulSeMaker [33]	FeelReal [38]
Temporal Sync. Paradigm		E	T	T	T	T	T	T	T	E & T	T
Authoring GUI Approach		T & S	W	T	T	T	T	T	T	W	T
Sync. of Multiple Media and SE		✓								✓	
SE Group								✓			
SE Reuse								✓			
Spatial View Editing	**SE Position Editing**	✓		+/-			✓	✓	✓		+/-
	Rendering Editing	✓		✓	+/-	✓	✓	✓	✓	+/-	+/-
	SE Spatial View										
	Multimedia Spatial View Editing	✓									
Asynchronous Events	**Interactivity Support**	✓								+/-	
	Statement Assessment Support										
Simulator						✓			✓		
Ordinary User Support		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Template Support										✓	
Automatic extraction		✓			+/-				✓		
Publishing Formats		none	MPEG-V	none	MPEG-V	MPEG-V	MPEG-V	MPEG-V	MPEG-V	HTML5	none
SE Prefetching Support											
Error Analysis		✓									
Player						✓					✓
Feature Score		8	1	2.5	2	4	3	5	5	4	3

+/- (partially supported feature); ✓ (fully supported feature);
Temporal Synchronization Paradigm: E (event-based); T (timeline-based);
Authoring GUI Approach: T (temporal); S (spatial); W (wizard)

(1 point) and *Rendering Editing* (1 point). However, this tool partially supports *SE Position Editing*, which results in 0.5 points. H-Studio thus has 2.5 as *Feature Score*.

Most tools we discussed in this survey are based on the MPEG-V temporal synchronization paradigm. Therefore, they inherit the limitations of the timeline paradigm. One limitation refers to the lack of support for defining asynchronous events, such as user interaction and state variable assessments. Another limitation is the fact that those tools only support the authoring of sensory effect metadata (SEM files). They do not specify sensory effects in the context of mulsemedia applications. Consequently, these MPEG-V-based tools do not allow the definition of temporal relationships among nodes (traditional media or sensory effects). In contrast, only STEVE 2.0 and

MulseMaker allow the definition of nodes and temporal relationships to specify an application behavior and support the event-based temporal synchronization paradigm.

Furthermore, STEVE 2.0 is the only tool that offers a spatial GUI approach that provides spatial view editing for traditional multimedia and sensory effects. CrowdMuse and MulSeMaker are tools that use the wizard approach. However, CrowdMuse only provides a single interface with a video player and buttons to select the beginning and end time instants and intervals to annotate sensory effects with a single video, while MulseMaker gives a step-by-step window to guide authors through the authoring process. The remaining tools only provide a timeline view as an authoring GUI approach.

Figure 2 illustrates four quadrants defined according to temporal synchronization paradigm (event-based and timeline-based) and features to give an overview of the mulsemedia tools discussed. We placed the tools in the horizontal axis according to the *Feature Score* line in Table 2. This axis is divided in half in value 4.5. Thus, STEVE 2.0, which has the highest feature score (8), is most to the right, whilst CrowdMuse is leftmost with the lowest score.

The vertical axis classifies those tools into two groups. One of them defines the tools that are based on events to temporally synchronize mulsemedia documents. This first group contains STEVE 2.0 and MulSeMaker. Although MulSeMaker uses the timeline paradigm for continuous media, it also uses the event-based paradigm, as discussed in Section 4.9. The other group refers to tools that use the timeline-based synchronization paradigm. For example, SEVino has 4 of score and its temporal synchronization is based on timeline. We can remark that the majority of tools use the timeline-based paradigm and do not provide several desirable features for mulsemedia authoring tools listed in Section 5.

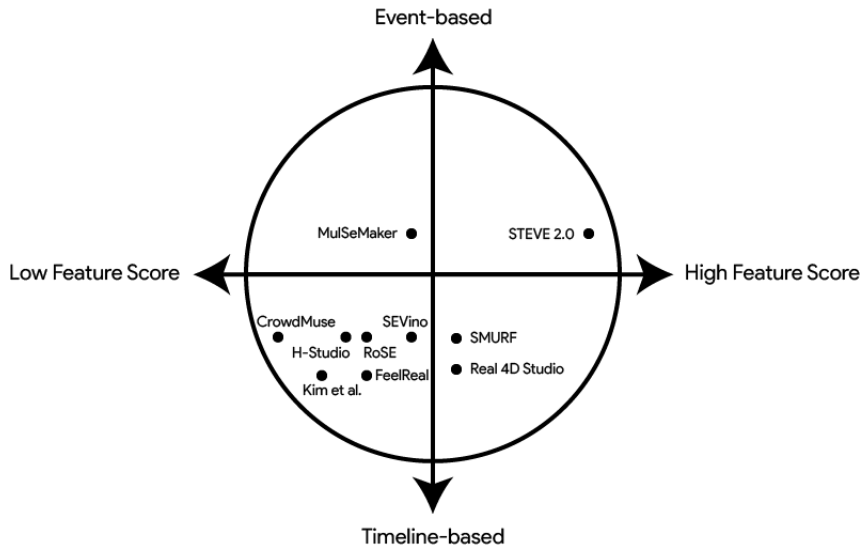

Fig. 2. Mulsemedia Tools Overview

7 MULSEMEDIA SIMULATORS AND PLAYERS

In order to run mulsemedia applications, several players have been proposed in the literature. Providing a player embedded into authoring tools is essential to enhance the whole process of deploying mulsemedia applications in the real world. Such players can present traditional

multimedia content and render sensory effects. To do this, these players not only communicate with multimedia displays but also with actuator devices sending them commands so that those devices can render sensory effects.

In addition to players, simulators have also been proposed. They use virtual actuators to support the author's effort of checking a mulsemedia application execution. To provide these virtual actuators, sensory effect simulators represent them using graphical symbols. Thus when an actuator should be activated, its graphical representation is highlighted to indicate that a sensory effect is being rendered by that actuator. In the following, some tools for playing and simulating mulsemedia applications are discussed.

7.1 SESim - Sensory Effect Simulator

Waltl et al. [87] proposed a simulator to aid the development of applications that make use of the MPEG-V standard to specify sensory effects. The simulator is called SESim, Sensory Effect Simulator. SESim receives a SEM file, which can be created using SEVino, and the audio/video (A/V) files as inputs. Then, the SESim XML parser extracts the sensory effects from that SEM file. Afterwards, those effects are forwarded to its simulator module. That module sends the A/V files to the player module and the extracted effects to the timer module. The timer module also receives the current playback time in order to activate/deactivate the corresponding virtual actuator.

7.2 SEMP - Sensory Effect Media Player

SEMP (Sensory Effect Media Player) is a mulsemedia application player that is also proposed in [87]. The player supports the following actuators for rendering sensory effects: amBX, Cyborg Gaming Lights and *Vortex Activ* systems [2]. The first one provides a wind effect through two fans with around 5000 rounds per minute (RPM). It also produces light effects using LEDs. Cyborg Gaming Lights also provide light effects. However, that system produces more powerful light effects than amBX. The Vortex Activ system provides scent effects. To do that, it consists of a set of four fans to spread scents in the environment. As SESim, SEMP should receive the audiovisual content and SEM files as inputs in order to render sensory effects. However, in SEMP, commands are forwarded to real devices for rendering sensory effects.

7.3 Sensible Media Simulator

Another simulator that is also compatible with MPEG-V is the Sensible Media Simulator [57]. This simulator was designed for taking advantage of car devices to produce sensory effects. The in-car entertainment system provides a graphical interface that shows the available car devices to be used as actuators. Users can also check the location of each device. As SESim, a SEM file and the audiovisual content are received as input. In addition, the capabilities of sensory devices in a car are described using CIDL (Control Information Description Language) of the MPEG-V standard. User sensory preferences are also specified with CIDL. The control information contained therein allows an application to be adapted to users and device capabilities. Additionally, the Sensible Media Simulator makes use of IIDL (Interaction Information Description Language) to describe sensed information and device commands.

7.4 Sensorama

Sensorama [20] is a player that is also able to receive SEM files as input. Additionally, it allows users to define an event list, in which an event represents a composition of sensory effects. For instance, an explosion can be an event that is composed by motion, temperature and wind effects. Sensorama's architecture is based on a 4D effect device control engine. This engine is responsible for loading and parsing SEM files, managing the synchronization between media and sensory

effects, mapping devices and data, and sending control signals to actuators. To achieve the goal of providing a 4D movie theater experience, the player uses a cave automatic virtual environment (CAVE). The CAVE system supports wind, light, fog, flash light and motion effects.

7.5 Multimedia Multisensorial 4D Platform

In [11], authors present a multimedia multisensorial 4D platform that focuses on delivering audio-visual content synchronized with only olfactory and thermal effects. The platform architecture is based on a central MCU (Microcontroller Unit) that represents the multisensory module. It is responsible for receiving sensory information from a server, performing parsing necessary to exchange data between the virtual and real worlds, and for sending commands to actuator devices, which render the olfactory and thermal sensory effects. To describe those commands and sensed information, the platform uses the MPEG-V standard. However, the high coupling of the MCU module implementation does not allow its extension to integrate new types of sensory effects and devices, and neither does it allow the reuse of the MCU with other mulsemedia applications.

7.6 Real 4D Studio Simulator

Real 4D Studio [77] is an authoring tool which provides an embedded simulator to allow authors to verify in a 3D virtual environment when sensory effects are activated during the mulsemedia application execution. The 3D simulation is performed by parsing SEM files written in MPEG-V and using the media control that is part of the Real4DEMaker module. The simulation presents 3D graphics for representing sensory effect actuators and also offers a media player together to show the synchronization between the audiovisual content (single video) and sensory effects. Since Real 4D Studio focuses on providing a 4D movie experience that includes armchair motion effects, the simulation also presents visual elements to represent each armchair movement.

7.7 PlaySEM SER 2

A mulsemedia framework, called PlaySEM SER 2, for dealing with heterogeneous applications and devices is proposed in [73]. The framework allows mulsemedia applications, whether timeline-based or event-based applications, to be rendered in the real world handling hardware configurations, communications issues and heterogeneous devices. Therefore, in the mulsemedia authoring phase, authors are able to focus only on the details of the mulsemedia application specification, such as the mulsemedia content and the spatio-temporal synchronization. To do that, this framework supports several communication protocols, metadada standards, connectivity interfaces and rendering devices. Previously in [74], PlaySEM SER did not provide a flexible architecture in which new protocols and devices could be added with no changes in its components. The first version of PlaySEM SER also only allowed the use of MPEG-V to describe the sensory effect metadata.

PlaySEM SER 2's architecture is divided in *Connectivity Interface*, S*ensory Effects Processing* and *Communication Broker*. The first is responsible for establishing connections with multiple and heterogeneous devices using a set of protocols, such as *Wi-Fi*, *Bluetooth* and *USB/Serial*. This module also provides a set of communication protocols, such as *CoAP*, *MQTT* and others to exchange messages between the devices and PlaySEM SER 2. The Sensory Effects Processing module reads metadata of sensory effects and converts them into messages to control the devices regardless of the specific type of device. This module allows mulsemedia systems to work not only with MPEG-V but also with other metadata types giving them flexibility. Concerning the Communication Broker module, it is a bridge between PlaySEM SER 2 and mulsemedia applications. In other words, this part conveys SEM descriptions to PlaySEM SER 2 and provides essential services for temporally synchronizing multimedia content with sensory effects such as starting, pausing and stopping. This communication is also made using different protocols such as UPnP, CoAP, MQTT, and WebSocket.

Although this proposal handles several issues concerning communication and connectivity with heterogeneous devices, it does not involve the specification of mulsemedia applications. For each new event or behaviour of the application, a software component must be implemented in a procedural language to recognize such an event or behaviour.

7.8 Josué et al.

Authors in [52] propose a sensory effect simulator to enable the preview of mulsemedia applications in a 3D room by receiving SEM files written in MPEG-V. This 3D environment consists of a center point and spread objects representing actuators. Users can organize physical objects, such as chairs and TV, and actuators using a WYSIWYG (What You See is What You Get) approach. They can also add and remove actuators. The direction of the actuators is always the center point of the room. The simulator uses colors to represent different sensory effect types and to indicate when actuators are active. The color intensity represents the effect intensity associated with the actuator. In this simulator, the authors propose an extension of the MPEG-V's spatial model to place actuators in the 3D room. This extended model uses a spherical coordinate system to overcome the MPEG-V's spatial model limitation of representing small variations in the position of sensory effects. Moreover, the MPEG-V's spatial model is not suitable for handling animations in which effect position changes over time. The simulator, however, supports both spatial models.

8 CONCLUSIONS

The demand for producing mulsemedia applications has encouraged several studies in the authoring phase, in particular regarding mulsemedia authoring tools. Therefore, this article focused on various proposals of mulsemedia authoring tools. We also presented studies concerning mulsemedia modeling since they support the development of authoring tools and are essential in representing the structure of mulsemedia documents. Additionally, our review of the literature included a multimedia background that discusses multimedia models and authoring tools to support our study in a mulsemedia context.

Indeed, we outlined desirable features for multimedia authoring tools that worked as a basis for us to identify important features for mulsemedia authoring tools. The set of mulsemedia features highlighted in this work are also supported by our analysis of several mulsemedia authoring tools.

In addition to those contributions to the mulsemedia community, the following sections identify gaps and future directions in the research and development of mulsemedia authoring tools.

8.1 Gaps of Authoring Tools

Based on our comparison of mulsemedia authoring tools in Section 5, we have identified gaps regarding the proposals of those authoring tools. All these gaps should be addressed in future proposals and investigated to advance research on mulsemedia authoring.

8.1.1 Temporal Synchronization Paradigm. Most tools are based on the timeline temporal synchronization paradigm, which, however, presents limitations, as discussed in Section 2.1. The event-based paradigm, which is more expressive, is rarely explored by the tools. In this review, only STEVE 2.0 [31] and MulSeMaker [33] use the event-based paradigm to temporally synchronize traditional multimedia and sensory effects.

8.1.2 Synchronization of Multiple Media Items. STEVE 2.0 and MulSeMaker are the only tools that support synchronization with multiple media items. In other words, those tools allow users to synchronize multiple sensory effects with several traditional multimedia items (audio, video, image and text) in time and space. On the other hand, the majority of tools presented in this survey support the synchronization of several sensory effects with only one video. Exploring this gap in

authoring tools can help to understand how users deal with several items at same time. This will also encourage the production of complex mulsemedia applications.

8.1.3 Sensory Effect Group and Reuse. Although MPEG-V defines groups of sensory effects, SMURF [55] is the unique proposal that allows users to define groups of effects and reuse frequently-used sensory effects/groups. Users can take advantage of this feature to define more complex combinations of sensory effects.

8.1.4 Sensory Effect Spatial View. None of the tools presented in this study provides an effect spatial view where authors can verify the spatial behavior of sensory effects and traditional multimedia items at the same time integrated with the temporal view.

8.1.5 Asynchronous Events. Asynchronous events is an important feature for creating interactive and dynamic mulsemedia applications. However, this feature is rarely provided by the tools. This feature includes the definition of interactivity relations and statement assessment tests.

8.1.6 Template Support. The template support is not also often explored by mulsemedia tools. Templates can accelerate the authoring phase by giving users predefined structures of mulsemedia applications. To provide those templates, the scientific community needs to investigate several mulsemedia scenarios in different application fields (learning, health, entertainment etc).

8.1.7 Sensory Effect Prefetching. The prefetching of sensory effects is not supported by any tool discussed in this study. However, this feature is fundamental for dealing with delays that actuators may have [53] while rendering sensory effects. For example, actuators to render scent effects need time to be effective. In other words, users may experience the scent effect rendered after a delay.

8.2 Future Directions

As future directions in the mulsemedia authoring phase, we highlight 360° mulsemedia applications [23, 26, 73]. In this context, users are immersed in omnidirectional videos (also known as 360° videos) that are also synchronized with sensory effects. Users have the freedom to control the view from a full spherical panorama. This new class of mulsemedia applications comes with new challenges and requirements for the authoring phase. Therefore, studies towards the integration of 360° videos in mulsemedia applications and analysis of how we can support those videos in mulsemedia authoring tools are essential. Indeed, virtual reality (VR) technologies have been explored for enhancing 360° mulsemedia authoring tools [22].

Another future direction that could be addressed in the context of authoring tools is that of crossmodal correspondences [61], which may affect our perceptual experiences. This concept refers to a compatibility effect between attributes or dimensions of a stimulus in different sensory modalities. In [27], a list of crossmodal correspondences is presented among smell, taste, touch, hearing and sight. For instance, authors describe the bouba/kiki effect experiment [81], which demonstrates that we can associate a shape with a specific sound. Accordingly, it is not inconceivable that mulsemedia authoring tools can leverage crossmodal concepts to create effective mulsemedia applications. Last but not least, mulsemedia authoring tools of the future would be greatly aided by the emergence of modeling languages targeting mulsemedia, as well as by mulsemedia simulators and players - all are research efforts needing to be addressed by the community.

ACKNOWLEDGMENTS

This work was partially supported by CAPES, CNPq and FAPERJ. The authors also thank CAPES PRINT Program for partially funding this work.

REFERENCES

[1] Raphael Abreu, Douglas Mattos, Joel AF dos Santos, Gheorghita Ghinea, and Débora C Muchaluat-Saade. 2021. Towards content-driven intelligent authoring of mulsemedia applications. *IEEE MultiMedia* 28, 1 (2021), 7–16. https://doi.org/10.1109/MMUL.2020.3011383

[2] Oluwakemi A Ademoye and Gheorghita Ghinea. 2013. Information recall task impact in olfaction-enhanced multimedia. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 3 (2013), 1–16. https://doi.org/10.1145/2487268.2487270

[3] Adobe. 2021. Adobe Director. http://adobe.com/br/products/director.html

[4] Adobe. 2021. Adobe Premiere. http://www.adobe.com/products/premiere.html

[5] Nero AG. 2021. Nero Video 2021. http://nero.com/ptb/products/nero-video/

[6] Anas Ali Alkasasbeh and Gheorghita Ghinea. 2020. Using olfactory media cues in e-learning - perspectives from an empirical investigation. *Multimedia Tools and Applications* 79, 27 (2020), 19265–19287. https://doi.org/10.1007/s11042-020-08763-3

[7] James F Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM* 26, 11 (1983), 832–843. https://doi.org/10.1145/182.358434

[8] Apple, Inc. 2021. Final Cut Pro X. http://apple.com/br/final-cut-pro/

[9] Y Ayers, A Cohen, DCA Bulterman, et al. 2001. Synchronized multimedia integration language (smil) 2.0. *W3C Recommendations* (2001). https://www.w3.org/TR/2001/REC-smil20-20010807/

[10] Brian Bailey, Joseph A Konstan, Robert Cooley, and Moses Dejong. 1998. Nsync-a toolkit for building interactive multimedia presentations. In *Proceedings of the sixth ACM international conference on Multimedia*. 257–266. https://doi.org/10.1145/290747.290779

[11] Simona Bartocci, Silvello Betti, Giuseppe Marcone, Marco Tabacchiera, Fabrizio Zanuccoli, and Armando Chiari. 2015. A novel multimedia-multisensorial 4D platform. In *2015 AEIT International Annual Conference (AEIT)*. IEEE, 1–6. https://doi.org/10.1109/AEIT.2015.7415215

[12] Elisa Bertino, Elena Ferrari, and Marco Stolf. 2000. MPGS: An interactive tool for the specification and generation of multimedia presentations. *IEEE Transactions on Knowledge and Data Engineering* 12, 1 (2000), 102–125. https://doi.org/10.1109/69.842254

[13] Blackmagic. 2021. Davinci Resolve 17. http://blackmagicdesign.com/products/davinciresolve

[14] G. Blakowski and R. Steinmetz. 1996. A media synchronization survey: Reference model, specification, and case studies. *IEEE journal on selected areas in communications* 14, 1 (1996), 5–35. https://doi.org/10.1109/49.481691

[15] Susanne Boll and Wolfgang Klas. 2001. Z/sub Y/Xa multimedia document model for reuse and adaptation of multimedia content. *IEEE transactions on knowledge and data engineering* 13, 3 (2001), 361–382. https://doi.org/10.1109/69.929895

[16] Samia Bouyakoub and Abdelkader Belkhir. 2011. SMIL builder: An incremental authoring tool for SMIL Documents. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 7, 1 (2011), 1–30. https://doi.org/10.1145/1870121.1870123

[17] M Cecelia Buchanan and Polle T Zellweger. 1992. Specifying temporal behavior in hypermedia documents. In *Proceedings of the ACM conference on Hypertext*. 262–271. https://doi.org/10.1145/168466.171513

[18] Dick CA Bulterman and Lynda Hardman. 2005. Structured multimedia authoring. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 1, 1 (2005), 89–109. https://doi.org/10.1145/1047936.1047943

[19] Dick CA Bulterman, Lynda Hardman, Jack Jansen, K Sjoerd Mullender, and Lloyd Rutledge. 1998. GRiNS: A GRaphical INterface for creating and playing SMIL documents. *Computer Networks and ISDN systems* 30, 1-7 (1998), 519–529. https://doi.org/10.1016/S0169-7552(98)00128-7

[20] H. Y. Cho. 2010. Event-Based control of 4D effects using MPEG RoSE. In *Master's thesis - School of Mechanical, Aerospace and S. Engineering. Korea Adv. Inst. of Science and Technology*. https://koasas.kaist.ac.kr/handle/10203/45825

[21] Bumsuk Choi, Eun-Seo Lee, and Kyoungro Yoon. 2011. Streaming media with sensory effect. In *2011 International Conference on Information Science and Applications*. IEEE, 1–6. https://doi.org/10.1109/ICISA.2011.5772390

[22] Hugo Coelho, Miguel Melo, José Martins, and Maximino Bessa. 2019. Collaborative immersive authoring tool for real-time creation of multisensory VR experiences. *Multimedia Tools and Applications* 78, 14 (2019), 19473–19493. https://doi.org/10.1007/s11042-019-7309-x

[23] Ioan-Sorin Comşa, Estêvão Bissoli Saleme, Alexandra Covaci, Gebremariam Mesfin Assres, Ramona Trestian, Celso AS Santos, and Gheorghita Ghinea. 2019. Do I Smell Coffee? The Tale of a 360° Mulsemedia Experience. *IEEE MultiMedia* 27, 1 (2019), 27–36. https://doi.org/10.1109/MMUL.2019.2954405

[24] W3C World Wide Web Consortium. 2008. Synchronized Multimedia Integration Language. http://w3.org/TR/SMIL/

[25] W3C World Wide Web Consortium. 2014. HTML5. http://w3.org/TR/html5/

[26] Alexandra Covaci, Ramona Trestian, Estêvão Bissoli Saleme, Ioan-Sorin Comsa, Gebremariam Assres, Celso AS Santos, and Gheorghita Ghinea. 2019. 360° Mulsemedia: a way to improve subjective QoE in 360° videos. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2378–2386. https://doi.org/10.1145/3343031.3350954

[27] Alexandra Covaci, Longhao Zou, Irina Tal, Gabriel-Miro Muntean, and Gheorghita Ghinea. 2018. Is multimedia multisensorial ? - a review of mulsemedia systems. *ACM Computing Surveys (CSUR)* 51, 5 (2018), 91. https://doi.org/10.1145/3233774

[28] Fabien Danieau, Jérémie Bernon, Julien Fleureau, Philippe Guillotel, Nicolas Mollet, Marc Christie, and Anatole Lécuyer. 2013. H-Studio: an authoring tool for adding haptic and motion effects to audiovisual content. In *Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology*. 83–84. https://doi.org/10.1145/2508468.2514721

[29] Marcello Novaes de Amorim, Estêvão Bissoli Saleme, Fábio Ribeiro de Assis Neto, Celso AS Santos, and Gheorghita Ghinea. 2019. Crowdsourcing authoring of sensory effects on videos. *Multimedia Tools and Applications* 78, 14 (2019), 19201–19227. https://doi.org/10.1007/s11042-019-7312-2

[30] Douglas Paulo de Mattos and Débora C Muchaluat-Saade. 2018. STEVE: a Hypermedia Authoring Tool based on the Simple Interactive Multimedia Model. In *Proceedings of the ACM Symposium on Document Engineering 2018*. ACM, 1–10. https://doi.org/10.1145/3209280.3209521

[31] Douglas P de Mattos, Débora C Muchaluat-Saade, and Gheorghita Ghinea. 2020. An approach for authoring mulsemedia documents based on events. In *2020 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 273–277. https://doi.org/10.1109/ICNC47757.2020.9049485

[32] Maria Cristina Ferreira De Oliveira, Marcelo Augusto Santos Turine, and Paulo Cesar Masiero. 2001. A statechart-based model for hypermedia applications. *ACM Transactions on Information Systems (TOIS)* 19, 1 (2001), 28–52. https://doi.org/10.1145/366836.366869

[33] M. F. de Sousa, C. A. G. Ferraz, R. Kulesza, I. Ayres, and M. Lima. 2017. MulSeMaker: An MDD Tool for MulSeMedia Web Application Development. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*. ACM, 317–324. https://doi.org/10.1145/3126858.3126883

[34] Romain Deltour, Nabil Layaida, and Daniel Weck. 2005. LimSee2: A Cross-Platform SMIL2. 0 Authoring Tool. *The European Research Consortium for Informatics and Mathematics–ERCIM News* 62 (2005). https://www.ercim.eu/publication/Ercim_News/enw62/deltour.html

[35] Paloma Díaz, Ignacio Aedo, and Fivos Panetsos. 2001. Modeling the dynamic behavior of hypermedia applications. *IEEE Transactions on Software Engineering* 27, 6 (2001), 550–572. https://doi.org/10.1109/32.926176

[36] Joel André Ferreira dos Santos and Débora Christina Muchaluat-Saade. 2012. XTemplate 3.0: spatio-temporal semantics and structure reuse for hypermedia compositions. *Multimedia Tools and Applications* 61, 3 (2012), 645–673. https://doi.org/10.1007/s11042-011-0732-2

[37] Marica Echiffre, Claudio Marchisio, Pietro Marchisio, Paolo Panicciari, and Silvia Del Rossi. 1998. MHEC-5-aims, concepts, and implementation issues. *IEEE MultiMedia* 5, 1 (1998), 84–91. https://doi.org/10.1109/93.664745

[38] FeelReal, Inc. 2021. FeelReal. https://feelreal.com

[39] International Organization for Standardization. 2015. Information technology – Coding of audio-visual objects – Part 11: Scene description and application engine. https://www.iso.org/standard/63548.html ISO/IEC 14496-11:2015.

[40] Kazutoshi Fujikawa, Shinji Shimojo, Toshio Matsuura, Shojiro Nishio, and Hideo Miyahara. 1991. Multimedia Presentation System\Harmony" with Temporal and Active Mediay 0. (1991). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.3918

[41] Richard Furuta and P David Stotts. 2001. Trellis: A Formally Defined Hypertextual Basis for Integrating Task and Information. *Coordination theory and collaboration technology* (2001), 341. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.6954

[42] Ombretta Gaggi and Augusto Celentano. 2002. A visual authoring environment for prototyping multimedia presentations. In *Fourth International Symposium on Multimedia Software Engineering, 2002. Proceedings.* IEEE, 206–213. https://doi.org/10.1109/MMSE.2002.1181614

[43] Ghinea, Timmerer, Lin, and Gulliver. 2014. Mulsemedia: State of the art, perspectives, and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications* 11, 1s (2014), 17. https://doi.org/10.1145/2617994

[44] W3C SYMM Working Group. 2012. https://www.w3.org/AudioVideo/

[45] Katharina Gsöllpointner, Ruth Schnell, and Romana Karla Schuler. 2016. *Digital Synesthesia: A Model for the Aesthetics of Digital Art.* Walter de Gruyter GmbH & Co KG.

[46] A. L. V. Guedes, R. G. de Albuquerque Azevedo, and S. D. J. Barbosa. 2017. Extending multimedia languages to support multimodal user interactions. *Multimedia Tools and Applications* 76 (2017), 5691–5720. https://doi.org/10.1007/s11042-016-3846-8

[47] R. L. Guimarães, R. M. de Resende Costa, and L. F. G. Soares. 2008. Composer: Authoring tool for iTV programs. In *European Conference on Interactive Television*. Springer, 61–71. https://doi.org/10.1007/978-3-540-69478-6_7

[48] Hazel Lynda Hardman et al. 1998. *Modelling and Authoring Hypermedia Documents.* Ph.D. Thesis. Univ. Amsterdam. http://www.cwi.nl/~lynda/thesis/

[49] Lynda Hardman and Dick CA Bulterman. 1995. Authoring support for durable interactive multimedia presentations. *STAR Report in Eurographics* 95 (1995). https://homepages.cwi.nl/~dcab/PDF/eg95.pdf

[50] Andrew Hunt and Scott (editors) McGlashan. 2004. Speech recognition grammar specification. https://www.w3.org/TR/speech-grammar/

[51] Jack Jansen and Dick CA Bulterman. 2009. SMIL State: an architecture and implementation for adaptive time-based web applications. *Multimedia Tools and Applications* 43, 3 (2009), 203–224. https://doi.org/10.1007/s11042-009-0270-3

[52] M. Josué, R. Abreu, F. Barreto, D. Mattos, G. Amorim, J. dos Santos, and D. Muchaluat-Saade. 2018. Modeling sensory effects as first-class entities in multimedia applications. In *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, 225–236. https://doi.org/10.1145/3204949.3204967

[53] Marina Josué, Marcelo Moreno, and Débora Muchaluat-Saade. 2019. Mulsemedia preparation: a new event type for preparing media object presentation and sensory effect rendering. In *Proceedings of the 10th ACM Multimedia Systems Conference*. 110–120. https://doi.org/10.1145/3304109.3306230

[54] Muriel Jourdan, Nabil Layaïda, Cécile Roisin, Loay Sabry-Ismaïl, and Laurent Tardif. 1998. Madeus, and authoring environment for interactive multimedia documents. In *Proceedings of the sixth ACM international conference on Multimedia*. 267–272. https://doi.org/10.1145/290747.290780

[55] S. Kim. 2013. Authoring multisensorial content. *Signal Processing: Image Communication* 28, 2 (2013). https://doi.org/10.1016/j.image.2012.10.011

[56] S.K. Kim and J.J. Han. 2014. Text of white paper on MPEG-V. In *MPEG Group Meeting, ISO/IEC JTC*, Vol. 1. https://mpeg.chiariglione.org/standards/mpeg-v/architecture/white-paper-mpeg-v

[57] Sang-Kyun Kim, Yong-Soo Joo, and YoungMi Lee. 2013. Sensible media simulation in an automobile application and human responses to sensory effects. *ETRI Journal* 35, 6 (2013), 1001–1010. https://doi.org/10.4218/etrij.13.2013.0038

[58] Sang-Kyun Kim, Seung-Jun Yang, Chung Hyun Ahn, and Yong Soo Joo. 2014. Sensorial information extraction and mapping to generate temperature sensory effects. *ETRI Journal* 36, 2 (2014), 224–231. https://doi.org/10.4218/etrij.14.2113.0065

[59] Claudio Marchisio and Pietro Marchisio. 2001. Mediatouch: A native authoring tool for mheg-5 applications. *Multimedia Tools and Applications* 14, 1 (2001), 5–22. https://doi.org/10.1023/A:1011307421707

[60] Britta Meixner. 2017. Hypervideos and interactive multimedia presentations. *ACM Computing Surveys (CSUR)* 50, 1 (2017), 1–34. https://doi.org/10.1145/3038925

[61] Gebremariam Mesfin, Nadia Hussain, Elahe Kani-Zabihi, Alexandra Covaci, Estêvão B Saleme, and Gheorghita Ghinea. 2020. QoE of cross-modally mapped Mulsemedia: an assessment using eye gaze and heart rate. *Multimedia Tools and Applications* 79, 11 (2020), 7987–8009. https://doi.org/10.1007/s11042-019-08473-5

[62] Gebremariam Mesfin, Estevao Bissoli Saleme, Oluwakemi Ademoye, Elahe Kani-Zabihi, Celso Santos, and Gheorghita Ghinea. 2020. Less is (just as good as) more-an investigation of olfactory intensity and hedonic valence in mulsemedia QoE using heart rate and eye tracking. *IEEE Transactions on Multimedia* (2020). https://doi.org/10.1109/TMM.2020.2992948

[63] J. Monks, A. Olaru, I. Tal, and G. Muntean. 2017. Quality of experience assessment of 3D video synchronised with multisensorial media components. In *Broadband Multimedia Systems and Broadcasting (BMSB), 2017 IEEE International Symposium on*. IEEE, 1–6. https://doi.org/10.1109/BMSB.2017.7986129

[64] Débora Christina Muchaluat-Saade and Luiz Fernando Gomes Soares. 2002. XConnertor and XTemplate: improving the expressiveness and reuse in web authoring languages. *New review of hypermedia and multimedia* 8, 1 (2002), 139–169. https://doi.org/10.1080/13614560208914739

[65] Jin-Cheon Na and Richard Furuta. 2001. Dynamic documents: authoring, browsing, and analysis using a high-level petri net-based hypermedia system. In *Proceedings of the 2001 ACM Symposium on Document engineering*. ACM, 38–47. https://doi.org/10.1145/502187.502194

[66] Ryuichi Ogawa, Hiroaki Harada, and Asao Kaneko. 1990. Scenario-Based Hypermedia: A Model and a System. In *ECHT*, Vol. 90. 38–51. https://doi.org/10.1109/HICSS.1998.651682

[67] Douglas Paulo de Mattos, Júlia Varanda da Silva, and Débora Christina Muchaluat-Saade. 2013. NEXT: graphical editor for authoring NCL documents supporting composite templates. In *Proceedings of the 11th european conference on Interactive TV and video*. 89–98. https://doi.org/10.1145/2465958.2465964

[68] Fernando Pereira and Touradj Ebrahimi. 2002. *The MPEG-4 book*. Prentice-Hall.

[69] V Yu Peschanskii. 2020. Timewise data processing with programming language ASAMPL. *Scientific Notes of Taurida National V.I. Vernadsky University Serires: Technical Sciences* 1 (2020), 132–137. https://doi.org/10.32838/2663-5941/2020.1-1/24

[70] J. L. Peterson. 1981. *Petri Net Theory and Modeling of systems*. Prentice-Hall.

[71] B. Rainer, M. Waltl, E. Cheng, M. Shujau, C. Timmerer, S. Davis, I. Burnett, C. Ritz, and H. Hellwagner. 2012. Investigating the impact of sensory effects on the quality of experience and emotional response in web videos. In *Quality of Multimedia Experience, Fourth International Workshop on*. IEEE, 278–283. https://doi.org/10.1109/QoMEX.2012.6263842

[72] ITUT Rec. 2012. H. 761, Nested Context Language (NCL) and Ginga-NCL for IPTV Services, Geneva, Apr. 2009. https://www.itu.int/rec/T-REC-H.761

[73] Estêvão B Saleme, Alexandra Covaci, Gebremariam Mesfin, Celso AS Santos, and Gheorghita Ghinea. 2019. Mulsemedia DIY: a survey of devices and a tutorial for building your own mulsemedia environment. *ACM Computing Surveys (CSUR)* 52, 3 (2019), 1–29. https://doi.org/10.1145/3319853

[74] Celso A Saibel Santos, Almerindo N Rehem Neto, and Estevao B Saleme. 2015. An event driven approach for integrating multi-sensory effects to interactive environments. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 981–986. https://doi.org/10.1109/SMC.2015.178

[75] Joel AF Dos Santos, Débora C Muchaluat-Saade, Cécile Roisin, and Nabil Layaïda. 2018. A hybrid approach for spatio-temporal validation of declarative multimedia documents. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 4 (2018), 1–24. https://doi.org/10.1145/3267127

[76] Ansgar Scherp and Susanne Boll. 2005. Paving the last mile for multi-channel multimedia presentation generation. In *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International*. IEEE, 190–197. https://doi.org/10.1109/MMMC.2005.58

[77] Sang-Ho Shin, Keum-Sook Ha, Han-O Yun, and Yoon-Seok Nam. 2016. Realistic media authoring tool based on MPEG-V international standard. In *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 730–732. https://doi.org/10.1109/ICUFN.2016.7537133

[78] Esdras Caleb Oliveira Silva, Joel AF dos Santos, and Débora C Muchaluat-Saade. 2013. NCL4WEB: translating NCL applications to HTML5 web pages. In *Proceedings of the 2013 ACM symposium on Document engineering*. 253–262. https://doi.org/10.1145/2494266.2494273

[79] L. F. G. Soares and R. F. Rodrigues. 2005. Nested Context Model 3.0: Part 1-NCM Core. *Technical Report No. 18. Telemidia Lab, PUC-Rio, Rio de Janeiro* (2005). ftp://ftp.inf.puc-rio.br/pub/docs/techreports/05_18_soares-port.pdf

[80] Luiz Fernando G Soares, Rogério F Rodrigues, and Débora C Muchaluat Saade. 2000. Modeling, authoring and formatting hypermedia documents in the HyperProp system. *Multimedia systems* 8, 2 (2000), 118–134. https://doi.org/10.1007/s005300050155

[81] Charles Spence. 2011. Crossmodal correspondences: A tutorial review. *Attention, Perception, & Psychophysics* 73, 4 (2011), 971–995. https://doi.org/10.3758/s13414-010-0073-7

[82] Yevgeniya Sulema. 2017. ASAMPL: Programming language for mulsemedia data processing based on algebraic system of aggregates. In *Interactive Mobile Communication, Technologies and Learning*. Springer, 431–442. https://doi.org/10.1007/978-3-319-75175-7_43

[83] YS Sulema and VV Glinskii. 2020. Semantics and pragmatics of programming language ASAMPL. *Problems in Programming* 1 (2020), 74–83. https://doi.org/10.15407/pp2020.01.074

[84] Christian Timmerer. 2009. ISO/IEC CD 23005-3 3rd Edition Sensory Information. https://mpeg.chiariglione.org/standards/mpeg-v/sensory-information

[85] Guido Van Rossum, Jack Jansen, K Sjoerd Mullender, and Dick CA Bulterman. 1993. CMIFed: a presentation environment for portable hypermedia documents. In *Proceedings of the first ACM international conference on Multimedia*. 183–188. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.8969

[86] Thomas Wahl and Kurt Rothermel. 1994. Representing time in multimedia systems. In *ICMCS*. 538–543. https://doi.org/10.1109/MMCS.1994.292502

[87] M. Waltl, B. Rainer, C. Timmerer, and H. Hellwagner. 2013. An end-to-end tool chain for Sensory Experience based on MPEG-V. *Signal Processing: Image Communication* 28, 2 (2013), 136–150. https://doi.org/10.1016/j.image.2012.10.009

[88] Markus Waltl, Christian Timmerer, and Hermann Hellwagner. 2010. Improving the quality of multimedia experience through sensory effects. In *2010 Second International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 124–129. https://doi.org/10.1109/QOMEX.2010.5517704

[89] Roberto Willrich, Pierre de Saqui-Sannes, Patrick Sénac, France ENSICA, and Michel Diaz. 2000. HTSPN: an experience in formal modeling of multimedia applications coded in MHEG or Java. *Design and Management of Multimedia Information Systems: Opportunities and Challenges: Opportunities and Challenges* (2000), 380. https://doi.org/10.4018/978-1-930708-00-6.ch019

[90] Chun-Chuan Yang, Chen-Kuei Chu, and Yung-Chi Wang. 2008. Extension of Timeline-based Editing for Non-deterministic Temporal Behavior in SMIL2. 0 Authoring. *Journal of Information Science & Engineering* 24, 5 (2008). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.193.8844

[91] Z. Yuan, G. Ghinea, and G. Muntean. 2015. Beyond multimedia adaptation: Quality of experience-aware multi-sensorial media delivery. *IEEE Transactions on Multimedia* 17, 1 (2015), 104–117. https://doi.org/10.1109/TMM.2014.2371240