

How to integrate interactions into video editing software?

Céline Jost*

Paris 8 University, CHArt laboratory, celine.jost@univ-paris8.fr

Brigitte Le Pévédic

Southern Brittany University, LAB-STICC laboratory, brigitte.le-pevedic@univ-ubs.fr

This paper explores an idea to enrich existing mulsemmedia editing software, getting inspiration from music scores, to have them integrate actions coming from the viewer while editing a mulsemmedia. In this paper, we reflect on the possibility to do it with a unique timeline. Thus, we propose to completely change the point of view. Instead of trying to insert the interaction in the timeline, we propose to cut the media into several parts and insert them into the interaction.

CCS CONCEPTS • Human-centered computing~Human computer interaction (HCI)~Interactive systems and tools~User interface programming •Software and its engineering~Software notations and tools~Context specific languages~Visual languages

Additional Keywords and Phrases: interaction, interactive mulsemmedia, media editing software, scenagram

ACM Reference Format:

Céline Jost and Brigitte Le Pévédic. 2022. How to integrate interactions into video editing software?. In SensoryX'22: Workshop on Multisensory Experiences, together with IMX 2022: ACM International Conference on Interactive Media Experiences. June 22-24, 2022. Aveiro, Portugal.

1 INTRODUCTION

In the PRIM project [1], we ambition to build software that allows users to create some scenagrams [2] – interactive scenarios, putting a human in interaction with digital devices – without programming because it is too complex for many people [3], [4]. Such software would benefit numerous domains: cinema with interactive mulsemmedia (multiple sensory media) [5]; education with personalized exercises; health with cognitive or rehabilitation exercises; art with digital artwork; plays with interactions between an actor and digital device in an

* Published in accordance with the terms of the Creative Commons Attribution 4.0 International Public License (CC BY 4.0). Permission to reproduce or distribute this work, in part or in whole, verbatim, adapted, or remixed, is granted without fee, provided that the appropriate credits are given to the original work, not implying any endorsement by the authors or by SBC. 2022 Brazilian Computing Society

improvisational theater; research with experimental settings allowing to explore the impact between technology and human, etc.

Multimedia editing software such as PlaySEM [6] or STEVE [7] seems to be the most advanced in the domain and seems to be very easy to learn and manipulate [8], [9]. However, it is impossible to add interactions directly in PlaySEM. If one wants to take actions done by the viewer into account, it is necessary to analyze humans' actions by programming and to generate a media file for PlaySEM which will then read the media file. STEVE proposes to directly synchronize humans' actions on the audiovisual content or to define them in a separate timeline. In our project, we would like to explore a different way by investigating the possibility to have a single timeline. Section 2 explains the way to create multisensory media through a timeline. Then, section 3 presents the issue we would like to discuss in this paper: time and multiple timelines. Section 4 explores the possibility to get inspiration from music scores. Section 5 proposes an idea to integrate pieces of media in the interaction in analogy with music scores. Finally, section 6 discusses challenges and limits and then concludes this paper.

2 MULSEMEDIA EDITING SOFTWARE APPROACH

Multimedia editing software is based on a timeline that looks like Figure 1. The principle is simple. To synchronize a sensory effect to the audiovisual content, the user must place a shape (here a rectangle) which represents the sensory effect in the timeline. The left border of the shape has to be placed just below the time when this effect has to be triggered. It is possible to add as many sensory effects as desired. For example, in Figure 1, a light effect will start after 1s, a wind effect after 4s, etc.

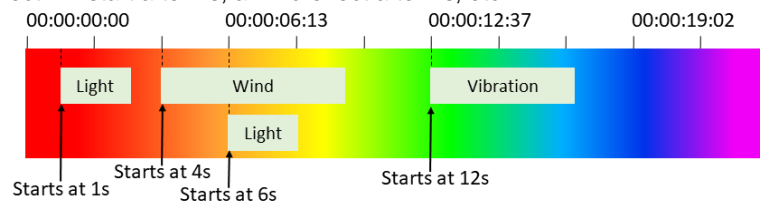


Figure 1: example of an editing software timeline with edition of sensory effects

To add interaction, there are two possibilities in the most advanced STEVE software [7], [10], [11], as illustrated in Figure 2. First, it is possible to place humans' input in the timeline. But that requires defining a time window when this input will be taken into account. If the user acts just before or just after, the system will not react. Second, it is possible to place an input in a separate timeline which is not synchronized on time. This input can occur anytime. But in that case, how is it possible to act on the audiovisual content?

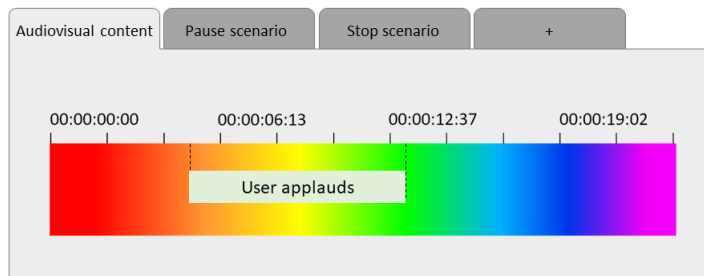


Figure 2: example of humans' action integration

3 MAIN ISSUE: TIMELINES AND TIME

One of the main differences between visual programming languages and multimedia editing software is time management. The first category is algorithmic-centered and based on an event temporality, meaning that each action occurs according to events (pressing a button, detecting a gesture, recognizing a sentence, ...), while the second one is media-centered and based on a real temporality. Figure 3.A shows an example of a program built on a timeline. This timeline indicates a relative time. First, the robot expresses the hello behavior. Then, nothing happens until the human answers with a hello gesture. If done, the robot asks a question. Each action and event are interpreted along the timeline, from left to right, without any notion of time.

The Figure 3.B shows an example of the timeline used in an editing software. In this example, a "Hello world!" subtitle is displayed at the beginning of the media. Then a video starts at 5s. Each timeline is synchronized on the same time. And we can see a vertical line around 14s that indicates the moment of the current playing.

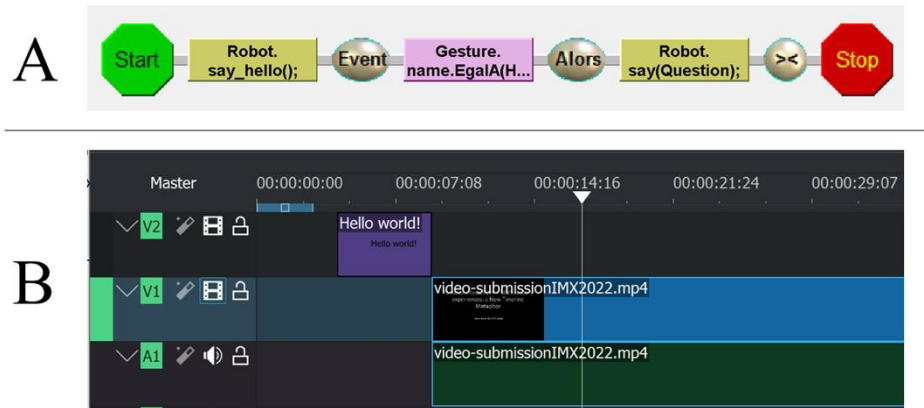


Figure 3: (A) example of a program built on a timeline, made with the AmbiProg software [12]
(B) example of a video editing area, made with the Kdenlive software [13]

Both of these pieces of software have a timeline. However, these timelines are based on a different time paradigm and are incompatible. That means it is impossible to integrate events, such as in Figure 3.A in a time-based timeline such as in Figure 3.B.

In this paper, we reflect on the possibility of using a single timeline with the aim to add events directly to the audio-visual content timeline. So, the question is: **is it and how is it possible to add events to the audio-visual content, in a unique timeline?**

4 MUSIC SCORES AS AN INSPIRATION?

Music may be a good starting point giving us a link between the current editing software and our goal in the PRIM project. First, music scores are of interest because they are timelines. However, they are a bit different because time is represented by coding, not by a time axis. For example, the whole-note, the half-note, and the quarter-note have the same visual size but last a different time, varying from one to four. However, these three notes take the same place on the timeline. Millions of musicians perfectly master and use this kind of timeline. Second, whatever the duration of the music is, music scores have line returns, which means there is no infinite vertical or horizontal scrolling such as in media editing software. This makes navigation easier than in computing

programs which can be very complex. Musicians are used to following the timeline from left to right with line returns and with several sheets of paper. It's interesting because it is possible to display a very long production, having easy navigation and reading.

Moreover, there is also a major point. Music scores already contain algorithmic: "sleep" (with break mark, caesura, rest, or multi-measure rest), iteration (with tremolo, repeat sign, simile mark, Da Capo, or Dal Segno), "go to" (with the coda sign), and even condition (with volta bracket which indicates alternative measures). All that indicates it is possible to have an algorithm represented on a timeline with line returns.

Figure 4 shows an example of goto, loop, condition, and sleep[†]. When playing, the musician interprets a kind of algorithm. The number 1 is a sign which indicates to the musician that she/he has to return to a specific time (goto). The number 2 is a condition because the first time the music after "To Coda" is playing, but not the second time. The number 3 is a loop. And the number 4 is a set of sleep.

However, even if music scores do not have a time axis, they are based on real temporality with a time that goes on and never stops, while event temporality uses a relative time where action may happen anytime. **How can we get inspiration from music scores while manipulating event time?**

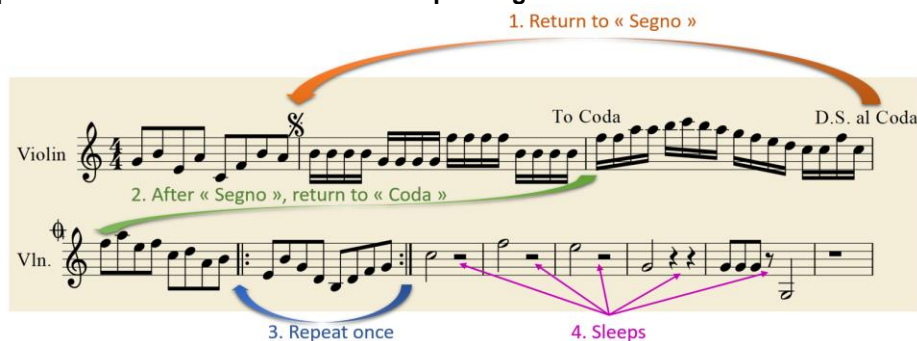


Figure 4: example of goto, loops, conditions and sleeps in a music sheet, made with the MuseScore software [14]

5 PROPOSAL

In the example given in Figure 2, the approach is to integrate the interaction into the audiovisual content. We propose to change the point of view and integrate the media into the interaction getting inspiration from music scores. To reach this objective, it is necessary to make the media interruptible such as a sequence of notes. A possibility is thus to cut the media into several parts as shown in Figure 5. Each part is like a music note, can be represented with the same graphical size but has a different duration. Within each part, it would be possible to pause and restart the media like if the musician stops playing or to take into account an anticipated human action that occurs at an expected time, but not to take a human action that can occur anywhen into account. Humans' actions would be taken into account only apart from each part, between each part.

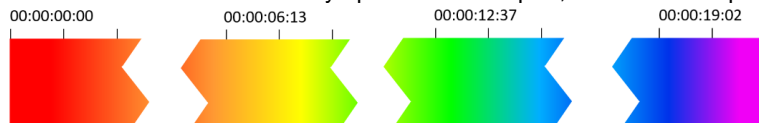


Figure 5: illustration of a media cut in several parts

[†] See an example of execution in <https://youtu.be/hF0gae7Ziw4>

After having cut the media into several parts, it is possible to visually represent the links between each part using a graph as shown in Figure 6. In our example, the media starts with part A. If an action is done by the user, part B can be played. And we can observe that there are two possibilities after part B: either part D which is the end of the media or part C which then goes to part A. In this graph, we can observe a condition formed by the possibility to have C or D after B, and a loop formed by circular links between A, B, and C.

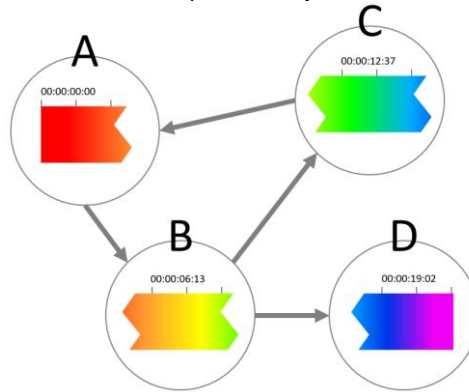


Figure 6: graph making links between several parts of media

Therefore, the challenge consists in presenting this graph in the form of a timeline. To represent the loop between A, B, and C, we can use two music symbols: the begin-repeat-bar and the end-repeat-bar. However, it is not really possible to use a music symbol to represent the condition with a Segno/Coda because D may happen or not. A music score describes a sequence (or a way). There can be alternatives, but each note has to be played at least once. Thus, if we flatten the graph, we can obtain at best the timeline in Figure 7 where is drawn an alternate timeline for D.

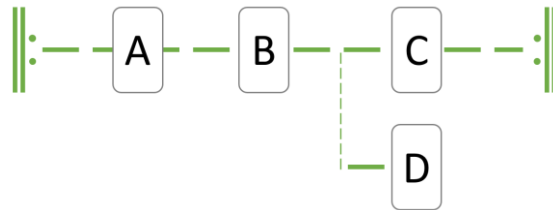


Figure 7: flattening of the graph into a timeline

Thus, for this little example, we obtain a main timeline with an optional timeline for B. Having an optional timeline can be a solution for a short-time alternative but not for representing another way of the story because we will have two different timelines. We can see here that, according to the graph, the timeline result can be terribly complex.

A second reflection can be to consider D as a jump toward a new timeline. Thus, we can imagine that in a system where there are multiple timelines, such as in STEVE, the main timeline (or graph) may define the links between all these timelines. More generally, we can imagine a system where it is possible to jump anywhen toward a piece of media as illustrated in Figure 8. If each part of media is numbered like each bar in a music score, it is then possible to jump toward any part of media according to its number or to another timeline. In the

figure, the alternative is represented by a circle with two outputs: an arrow pointing to element 1 of the current timeline and an arrow pointing to element 1 of the P2 timeline.

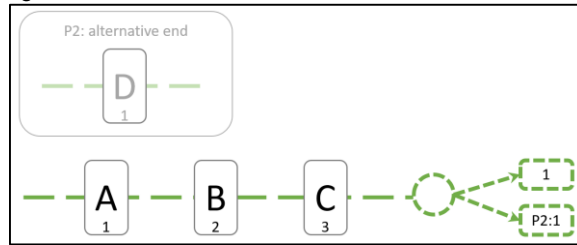


Figure 8: example of a timeline with jump toward other parts of the timeline or other timelines

Even if we can explore the possibility to reuse the approach used in music scores – where it is possible to have a few algorithmic (loops, conditions, goto, and sleep) on a timeline – we face a problem much more complex than music. The question is open.

6 DISCUSSION AND CONCLUSION

In this paper, we reflect on a new point of view to make programming accessible for many people who find it too complex, getting inspiration from mulsemmedia editing software. And, we propose to make an analogy with music scores and cut the media into several parts and integrate it into a timeline with the aim to make a representation of an interaction graph, compatible with media editing software. We propose to take inspiration from music scores to add symbols to the timeline to have a few algorithmic. But it is necessary to find a compromise because, even if music scores are easily interpreted by millions of musicians, they are based on music theory which can require years of learning. Moreover, music and alternative story (or more generally scenagrams) are two different issues with two different levels of difficulties. Thus, we have first to limit the number and the types of symbols to create, and second to find a way to avoid an alternate timeline.

In addition, we have to determine what is required for scenagrams by exploring more specifically interactive mulsemmedia, multisensory exercises, interactive artwork, improvisational plays, or Human-Machine evaluation. Do we need loops? Do we need conditions? Do we need to define multiple actions at the same time? We have to limit what a scenagram is in order to remain as simple as possible.

Whilst other approaches [10], [11], [15] target synchronization, sensory effects, and playing the mulsemmedia, what is proposed here is quite possibly complementary and could be integrated with such approaches into broader software that can manage interactions.

In the PRIM project, we have developed an incomplete software prototype [16] to evaluate a timeline where it is possible to add alternative timelines (in order to do several actions in parallel) and where each timeline is independent of time. Our objective was to use a timeline metaphor to represent a graph where a node is achieved because an action occurs or because the precedent action is finished, independently of time. A preliminary study, conducted with 50 participants, showed that the majority of participants were able to understand and use this new timeline.

Thus, the perspectives will focus on defining the best compromise between offering the most possibilities to the user and having software easy to use. The major challenge to face then will be to design, in collaboration with the end-users, the graphical symbols which will suit the best and which will answer the needs.

ACKNOWLEDGMENTS

This work was funded by several grants from Paris 8 University and by a doctoral grant from the CLI doctoral school. We thank all the members of the PRIM project and all the participants in the evaluation. And we also thank the students who joined us for an internship to participate in the reflection.

REFERENCES

- [1] C. Jost *et al.*, "PRIM Project: Playing and Recording with Interactivity and Multisensoriality," in *IMX 2021: ACM International Conference on Interactive Media Experiences*, 2021.
- [2] C. Jost, B. Le P ev edic, and G. Uzan, "Using Multisensory Technologies to Stimulate People: a Reflexive Paper on Scenagrams," in *SensoryX '21: Workshop on Multisensory Experiences, together with IMX 2021: ACM International Conference on Interactive Media Experiences*, 2021.
- [3] E. Coronado, F. Mastrogiovanni, B. Indurkha, and G. Venture, "Visual Programming Environments for End-User Development of intelligent and social robots, a systematic review," *J. Comput. Lang.*, vol. 58, p. 100970, Jun. 2020.
- [4] K. Howland and J. Good, "Learning to communicate computationally with Flip: A bi-modal programming language for game creation," *Comput. Educ.*, vol. 80, pp. 224–240, Jan. 2015.
- [5] G. Ghinea and O. Ademoye, "A user perspective of olfaction-enhanced mulsemedia," in *Proceedings of the International Conference on Management of Emergent Digital EcoSystems - MEDES '10*, 2010, p. 277.
- [6] E. B. Saleme and C. A. S. Santos, "PlaySEM: a Platform for Rendering MulSeMedia Compatible with MPEG-V," in *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web - WebMedia '15*, 2015, pp. 145–148.
- [7] D. P. De Mattos and D. C. Muchaluat-Saade, "Steve: A hypermedia authoring tool based on the simple interactive multimedia model," *Proc. ACM Symp. Doc. Eng. 2018, DocEng 2018*, Aug. 2018.
- [8] M. Waltl, C. Timmerer, and H. Hellwagner, "A test-bed for quality of multimedia experience evaluation of Sensory Effects," in *2009 International Workshop on Quality of Multimedia Experience*, 2009, pp. 145–150.
- [9] M. Waltl, B. Rainer, C. Timmerer, and H. Hellwagner, "An end-to-end tool chain for Sensory Experience based on MPEG-V," *Signal Process. Image Commun.*, vol. 28, no. 2, pp. 136–150, Feb. 2013.
- [10] R. S. De Abreu, D. Mattos, J. Dos Santos, G. Ghinea, and D. C. Muchaluat-Saade, "Toward Content-Driven Intelligent Authoring of Mulsemedia Applications," *IEEE Multimed.*, vol. 28, no. 1, pp. 7–16, Jan. 2021.
- [11] D. P. De Mattos, D. C. Muchaluat-Saade, and G. Ghinea, "An Approach for Authoring Mulsemedia Documents Based on Events," *2020 Int. Conf. Comput. Netw. Commun. ICNC 2020*, pp. 273–277, Feb. 2020.
- [12] C. Jost, B. Le P ev edic, and D. Duhaut, "ArCo: An architecture for children to control a set of robots," *Proc. - IEEE Int. Work. Robot Hum. Interact. Commun.*, pp. 158–164, 2012.
- [13] "Kdenlive - Video Editing Freedom." [Online]. Available: <https://kdenlive.org/fr/>. [Accessed: 02-Apr-2022].
- [14] D. Todea, "The Use of the MuseScore Software in Musical E-Learning," *Virtual Learn.*, p. 88, 2015.
- [15] D. P. De Mattos, C. Muchaluat-Saade, D ebora, and G. Ghinea, "Beyond Multimedia Authoring: On the Need for Mulsemedia Authoring Tools," *ACM Comput. Surv.*, vol. 54, no. 7, Jul. 2021.
- [16] C. Jost, B. Le P ev edic, J. Debloos, and G. Uzan, "Interactions in Multisensory Experiences: Toward a New Timeline Metaphor," in *SensoryX '22: Workshop on Multisensory Experiences, together with IMX 2022: ACM International Conference on Interactive Media Experiences*, 2022, p. 8.