

COMEDIANNOTATE: TOWARDS MORE USABLE MULTIMEDIA CONTENT ANNOTATION BY ADAPTING THE USER INTERFACE

Christian Frisson^(1,2), Sema Alaçam⁽³⁾, Emirhan Coşkun⁽³⁾, Dominik Ertl⁽⁴⁾,
Ceren Kayalar⁽⁵⁾, Lionel Lawson⁽¹⁾, Florian Lingenfeller⁽⁶⁾, Johannes Wagner⁽⁶⁾

⁽¹⁾ Communications and Remote Sensing (TELE) Lab, Université catholique de Louvain (UCLouvain), Belgium;

⁽²⁾ Circuit Theory and Signal Processing (TCTS) Lab, Université de Mons (UMons), Belgium;

⁽³⁾ Architectural Design Computing, Institute of Science and Technology, Istanbul Technical University (ITU), Turkey;

⁽⁴⁾ Institute of Computer Technology, Vienna University of Technology, Vienna, Austria;

⁽⁵⁾ Computer Graphics Lab (CGLab), Sabancı University, Istanbul, Turkey;

⁽⁶⁾ Lehrstuhl für Multimedia-Konzepte und Anwendungen (MM), Universität Augsburg, Germany

ABSTRACT

This project aims at improving the user experience regarding multimedia content annotation. We evaluated and compared current timeline-based annotation tools, so as to elicit user requirements. We address two issues: 1) adapting the user interface, by supporting more input modalities through a rapid prototyping tool and by offering alternative visualization techniques of temporal signals; and 2) covering more steps of the annotation workflow besides the task of annotation itself: notably recording multimodal signals.

We developed input devices components for the OpenInterface (OI) platform for rapid prototyping of multimodal interfaces: multitouch screen, jog wheels and pen-based solutions. We modified an annotation tool created with the Smart Sensor Integration (SSI) toolkit and componentized it in OI so as to bind its controls to different input devices. We produced mockups sketches towards a new design of an improved user interface for multimedia content annotation, and started developing a rough prototype using the Processing Development Environment.

Our solution allows to produce several prototypes by varying the interaction pipeline: changing input modalities and using either the initial GUI of the annotation tool, or the newly-designed one. We target usability testing to validate our solution and determine which input modalities combination best suits given use cases.

KEYWORDS

Multimodal annotation, rapid prototyping, information visualization, gestural interaction

1. INTRODUCTION

This project attempts to provide a tentative toolbox aimed at improving the user interface of current tools for multimedia content annotation. More precisely, this project consists in combining efforts gathered in fields such as rapid prototyping, information visualization, gestural interaction; by adding all the necessary and remaining components to a rapid prototyping tool that allows to visually program the application workflow, in order to refine the user experience, first of one chosen annotation tool. This toolbox is a first milestone in our research, a necessary step to undertake usability tests on specific scenarios and use cases after this workshop.

This report is structured as follows. In Section 2 we define the context and scope of the project, i.e. “multimedia content” (Section 2.1) “annotation” (Section 2.2) and list possible use cases

(Section 2.3) and testbeds (Section 2.4). In Section 3, we summarize the current problems of timeline-based multimedia content annotation tools, based on previous comparisons (Section 3.1) and on an evaluation we undertook during the workshop (Section 3.2), then we explain why we chose to adapt the SSI annotation tool (Section 3.3). In Section 4, we describe the method we opted for: through a user-centered approach (Section 4.1), we restricted our design to two modalities (Section 4.2): visualization (Section 4.2.1) and gestural input (Section 4.2.2), among other possible modalities (Section 4.2.3); we thus used a rapid prototyping (Section 4.3) visual programming tool (Section 4.3.1) for the user interface (Section 4.3.2), the OpenInterface platform (Section 4.3.2.2), and a rapid prototyping tool for visualization (Section 4.3.3), the Processing Development Environment (Section 4.3.3.2). In Section 5, we summarize our results: we proposed a new tentative design of an improved user interface (Section 5.1), illustrated with mockups (Section 5.1.2) and an early prototype (Section 5.1.3); and we developed components for the OpenInterface platform (Section 5.2) for gestural input modalities (Section 5.2.1), control of the SSI annotation tool (Section 5.2.2). In Section 6, we underline our future works: a more robust prototype integrated into the MediaCycle framework for multimedia content navigation by similarity (Section 6.1) and subsequent usability tests to validate our designs (Section 6.2). Finally, we conclude in Section 7.

2. CONTEXT: ANNOTATION OF MULTIMEDIA CONTENT

2.1. What do we mean by “multimedia”

“Multimedia data” commonly refers to content (audio, images, video, text...) recorded by sensors and manipulated by all sorts of end-users. In contrast, the term “multimodal data” describes signals that act as ways of communication between humans and machines. Multimodal data can be considered as of a subset of “multimedia data”, since the first are produced by human beings. Multimedia data thus broaches a wider range of content (natural phenomena, objects, etc...). Annotation tools help analyzing multimedia data, but also make use of multimodal signals within their user interface.

2.2. What do we mean by “annotation”

The following questions illustrate the issues we faced while understanding each others on a generic definition of the term “annotation”:

- Who is doing it? Human(s) and/or machine(s)?:
 - automatic annotation consists in extracting metadata using signal processing algorithms with no (or limited) parameter tweaking required from the user;
 - “manual” annotation is performed by humans adding metadata to data using various user interaction techniques;
 - semi-automatic annotation combines both approaches, sequenced in time. For instance: once data is loaded in the annotation tool, feature extraction algorithms run in the background on a subset, the user is then asked to correct these automated annotation, then a process propagates the corrections to the whole dataset.
- In case of humans, what about standard versus expert users? Is it being performed collaboratively by multiple users?
- What kind of data is annotated? “Multimedia content” and/or “multimodal signals”?
- When is it performed? Online and/or offline?
- For what purpose? Which use cases, scenarii?

Semiotically, from the user perspective and viewpoint, two types of annotation can be discriminated:

- *semantic*: words, concepts... that can be assorted in domain ontologies,
- *graphic*: baselines, peaks... with a tight relation to the gestural input required to produce them

Additionally, Kipp proposes a spatio-temporal taxonomy in [29]: *shape* (or geometric representation), *number* (of occurrences), *order* (chronological or not), *rigidity* and *interpolation* (discrete, linear or spline).

We opted for the following definition: annotations consist in “adding metadata to data in order to extract information”, that is contradictory with [42] which confronts “annotation” and “metadata”, the first term considered time-dependent by the author while the second isn’t.

2.3. Possible use cases

We had in mind to propose a toolbox with which the user can adapt the annotation tool to his/her needs, instead of having to use a different tool for each domain of use, for instance: corpora archival, multimedia library sorting, sensor recordings analysis, etc.. There are numerous possible uses of multimedia content annotation, here follows a subset applied to multimedia arts:

- annotation of motion capture [23], for instance with online errors notification while recording for offline reconstruction of missing data;
- analysis of dancers’ performances [56] requiring diverse types of sensors, training mappings of gesture-based dancers interfaces using performances recordings [16, 19];
- preparation of material for live cinema performances [37];
- multimedia archival and restoration [49]...

2.4. Possible testbeds

We tried to adapt the project scope to fit it better to some MSc/PhD participants topics, by considering two more testbeds besides timeline-based annotation tools.

2.4.1. Timeline-based Annotation Tools

These tools focus on the analysis of temporal signals or time-series and offer a great challenge regarding handling time for navigation and annotation purposes. It has to be noted that most participants already had some experience with multimedia edition tools, requiring similar navigation methods and offering a subset of the variety of possible annotations.

2.4.2. Multimedia Collection Managers

These tools, such as iTunes for music libraries, share design questions with Emirhan’s MSc (2D visualization and representation of massive datasets, in his case in the context of social networks) and Christian’s PhD (similar, applied to multimedia content). We discarded this testbed because we haven’t found any already-existing opensource tool that would offer flexible annotation further than basic metadata management (ID3 tags for music, movie “credits” information, etc...) for audio and video media types (however, we found some for image or text).

2.4.3. Panoramic Image based 3D Viewers and VR World Viewers

These tools, such as Google Earth, HDView Gigapixel Panoramas [32] and Photosynth [55], were particularly interesting regarding Ceren’s PhD work [27]. We discarded this testbed because developing a simple 3D viewer with annotation support or even integrating navigation and annotation through Google Earth API would have taken too much time, leaving not much time to deal with real research issues (for example: occlusion-free tag 3D position considering a variable user viewpoint).

3. TIMELINE-BASED MULTIMEDIA CONTENT ANNOTATION TOOLS: FROM PROBLEMS, TOWARDS USER REQUIREMENTS

3.1. Summary of current problems

Plenty of pre-existing works compared annotation tools and elicited emerging requirements, for instance throughout the last decade [6, 8, 12, 48, 50]. Based on these observations and readings, we summarize the following issues regarding how annotation tools could be improved (checked boxes emphasizing the ones we planned to address throughout the workshop):

- multimedia: better file formats support [6, 50], time-based media other than audio and video [6];
- scale: number and/or length of media elements in the database;
- reusability: toolboxes/frameworks rather than isolated tools specific to a given context of use [12, 48], portability over multiple operating systems [8, 50];
- accessibility: client/server applications rather than desktop applications working with local media databases;
- interactivity: a multimodal user interface could help enhance the pleasurability and efficiency of these tools that are generally WIMP-based [6, 12, 48], so as to provide a single used interface that allows:
 1. to monitor signal feeds while recording datasets,
 2. optionally to add annotations while recording,

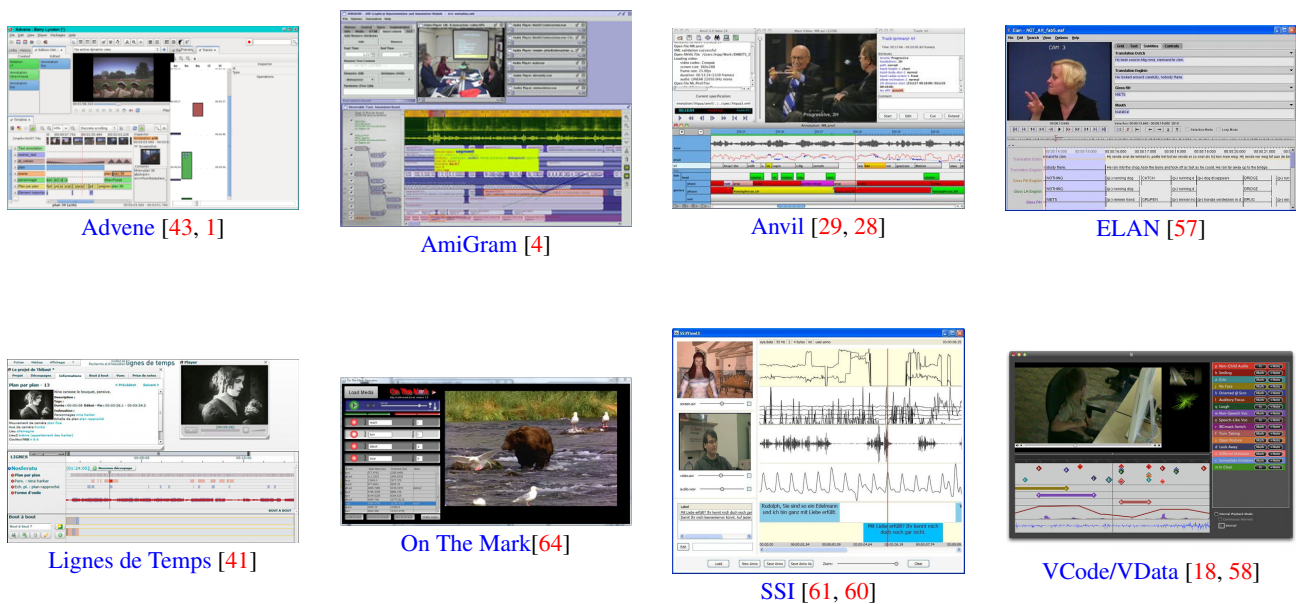


Figure 1: Screenshots of our selection among the available annotation tools. Image copyrights remain with their authors.

- 3. to edit or correct annotations;
- 4. a more natural, usable, pleasurable user interface (pen and touch).
- workflow: supporting of the full annotation workflow [12, 18]:
 1. one administrator prepares (design of a template and choice of coders);
 2. several coders record;
 3. several coders annotate;
 4. the administrator analyses results (coder agreement...).

3.2. Evaluation and testing during eNTERFACE'10

We tested 8 opensource or free tools, with screenshots in Fig 1, at least with one participant assigned to each (practically, two participants tested each), alphabetically: Advene [43, 1], AmiGram, Anvil [29, 28], ELAN [57], Lignes de Temps [41], On The Mark [64], Smart Sensor Integration (SSI) [61, 60] and VCode/VData [18, 58]; so as to better understand the concerns with a hands-on approach.

We produced detailed comparison in 3 tables that are available online on the eNTERFACE'10 wiki¹, focusing on:

1. development criteria (quantitative): OS, licence, development languages, supported formats...;
2. context, usage (quantitative): media types, scope, field of use...;
3. eNTERFACE participants feedback (qualitative): subjective comments on usability and pleasurability raised by the participants while testing these tools.

¹<http://enterface10.science.uva.nl/wiki/index.php/CoMediAnnotate:Framework:Annotation:Tools>

A first round of selection based on development considerations (operating system, development language and licenses) narrowed down the choice among 3 candidates out of the 8 tested: AmiGram, ELAN and SSI.

- implementation: in C++ or Java or C# or Python, supported by the rapid prototyping platform for multimodal interfaces we chose (as explained in Section 4.3.2.2);
- license: necessarily open-source so that we could modify the source code;
- compatibility: running on most operating systems possible, the common denominator operating system among participants being Windows.

3.3. Chosen tool for adaptation: Smart Sensor Integration (SSI)

3.3.1. Description

The SSI toolkit [61, 62] developed within the CALLAS EU project by two of the participants, Johannes and Florian, is a framework for multimodal signal processing in real-time. It allows the recording and processing of human generated signals in pipelines based on filter and feature extraction blocks. By connecting a pipeline with a classifier it becomes possible to set up an online recognition system. The training of a recognition model requires the collection of a sufficient number of samples. This is usually accomplished in two steps: 1) setting up an experiment to induce the desired user behavior, 2) review the recorded signals and add annotation to describe the observed behavior. For this purpose SSI offers a an annotation tool for multimedia signals. Signals recorded with SSI can be reviewed and annotated within this tool (see Fig. 2).

Depending on the length of the recordings (usually several hours) annotation can turn out to be an extremely time-consuming task. Currently the tool is controlled via simple mouse and keyboard commands. This is not always the fastest way and after some while of continuous use can become inconvenient for the user.

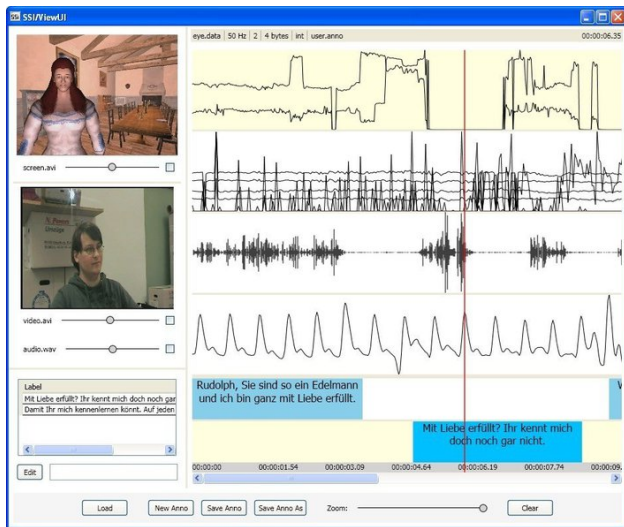


Figure 2: In SSI recorded sessions are visualized together with annotation tracks that describe the observed user behavior. The screenshot shows four signals (top down: eyegaze, head tracking, audio and blood volume pulse) and two annotation tracks (here: the transcription of the dialog between the virtual character and the user). On the left side videos of the application and the user are displayed. Screenshot from [62].

Hence, the tool would greatly benefit from alternative ways of interaction, such as Nintendo’s WiiRemote control or a gamepad.

3.3.2. Reasons for the choice

- We are in close contact with its developers who participated to the project during the first week.
- The core is separated from the UI.
- The simple annotation GUI is lightweight, hence simple to understand, and easy to replace.
- The toolkit not only proposes a simple annotation tool, but also feature extraction algorithms for automatic annotation, and could bridge the gap between multimedia content and multimodal signals annotation. This is of interest for some participants like Dominik for future works around adaptive multimodal interfaces by training [51].
- The development languages are compatible with the chosen rapid prototyping platform (see Section 4.3.2.2).

4. METHOD

4.1. User-centered approach

We opted for a user-centered approach [12] to conduct our research:

- in addition to gathering scientific documentation, we undertook a small contextual inquiry with eNTERFACE participants that had already had to use an annotation tool;
- before diving into software development, we cycled through and brainstormed on different design propositions using paper mockups;

- we produced a fast software and hardware prototype with off-the-shelf devices using rapid prototyping tools, as a first proof-of-concept, before rethinking the prototype with a more dedicated but slower to implement solution.

4.2. Two modalities of interest

Currently, we target standard experts (ie not “disabled” users such as blind people), yet such cases could be addressed since we are making use of a rapid prototyping tool for multimodal user interfaces. Ever since before computerized systems, two modalities were deeply rooted in the task of annotation: visualization and gestural input.

4.2.1. Visualization

The earlier visualization techniques regarding annotation were often offered by the recording device itself: sensor plots, video films, audio tapes, and so on... The closest task to multimedia content annotation is multimedia edition, notably with audio and video sequencers that can record signals, segment them, apply effects on them and realign them along the timeline.

Lots of techniques dedicated to time series have been proposed so far [3, 33]. Less standard information visualization techniques considering the user perception [63] might improve the task of multimodal annotation, during monitoring of recording processes and post-recording analysis. For a more in-depth analysis, different types of plots can help reduce the complexity of multidimensional data spaces and allow visual data mining. Animations between visualization techniques switched during the task may arouse cognitive effects and improve the user’s comprehension of the underlying information present within the displayed data [22, 5]. We follow this overview with specificities to some media types we chose to investigate: audio and video.

4.2.1.1. Audio: waveforms...

A survey of waveforms visualization techniques is proposed in [17], using visual variables to display more information than envelope or amplitude, rather: segments, frequency and timbral content, etc... Some advice is offered on how to visualize waveforms under small scale constraints, particularly by neglecting the negative part of the waveform or subtracting it to the positive part so as to overlap both, similar to a half-rectified signal. A regressive variation on these “mirrored graphs” called “n-band horizon graphs” [21], effectively reducing the height of time-series while keeping readability of information at high zoom factors, seems particularly useful for multitrack timeline representations.

4.2.1.2. Video: keyframes...

Video content is often represented by its frames or keyframes in various ways:

- all frames aligned in time horizontally;
- a subset of these sequenced in time and overlapped in location (such “animated GIF” image files serving as thumbnails on video hosting portals such as Archive.org);
- a standard video player where all frames are displayed on the same location, overlapped in time.

Other spatio-temporal content-specific techniques have been for video signals, for instance “MotionGrams” [26] or “slit/video scanning” [40], particularly suited to videos featuring movement

of recurring elements in the scene (again, for instance, dancers videos, among other examples in interactive arts [40]). Lee et al. proposed several keyframe browsing prototypes [36] characterized along a 3D design space: *layeredness* (single/multiple layer with/without links), *temporal orientation* (relative/absolute/none) and *spatial vs. temporal visualization*.

4.2.2. Gestural input

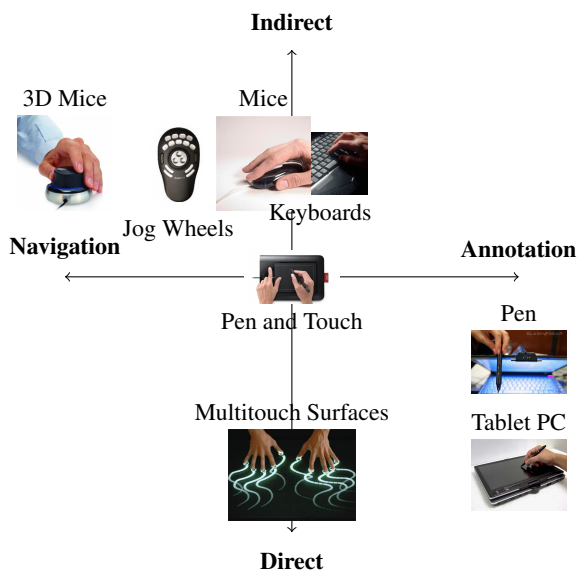


Figure 3: A selection of devices sorted on a 2-D space, indicating: horizontally whether each device seem suitable for navigation and/or annotation tasks, vertically whether the tied gestural input vs visual output modalities relation is direct or indirect.

Keyboards and mice interaction is still standard for most desktop applications [39], key bindings appears to be the fastest way of triggering momentary or ranged annotation when navigating on the signals with a constant playback speed [18]. Pen have been used by human people to annotate graphics and plots long before their recent computerized versions, now free-form [30] with styluses [2]. Jog wheels for navigating in audio and video signals have been widely used by experts of audio edition and video montage before multimodal annotation. Multitouch interfaces allow the combination of both navigation and annotation modes using one single gestural input modality. The direct or indirect gestural vs visual relation of the user interface can affect the spatial accuracy and speed of annotation tasks [52]. We have illustrated these concepts in Fig. 3 by representing gestural input modalities illustrated with common associated low-cost controllers.

4.2.3. Other possible modalities

As raised in Section 2.1, similar sensors can be used to record both the multimedia signals being annotated with the annotation tool and the multimodal signals used in the user interface from the tool, thus such modalities used for multimodal emotion recognition [61], for instance eye gaze could be used to improve the location of annotations and predict regions of interest for the user so as to better layout notifications; while voice input with speech

recognition could help produce instant user-defined tags or accurate dubbing of meeting recordings.

4.3. Rapid Prototyping

4.3.1. Scripted/textual versus visual programming

Signal processing and engineering specialists often use scripted/textual programming for their prototypes (for instance using Matlab) and they optionally switch to visual programming dataflow environments when realtime prototyping is of concern (with LabVIEW, Simulink, etc...). We believe that blending both approaches is convenient for the process of designing and prototyping the multimodal user interface of our adapted tool: visual programming gives a visual representation by itself of the underlying interaction pipeline, quite practical for exchanging design cues, while textual programming is quicker a designing simple and fast procedural loops, among other advantages.

4.3.2. Visual Programming Environments for Multimodal Interfaces

4.3.2.1. Existing visual programming tools

The number of multimodal prototyping tools and frameworks, dedicated to gestural input or generic towards most multimodal interfaces, has been increasing over the last two decades, yet none of them has been accepted so far as an industry standard. Among the vast availability, we would like to cite some that are still accessible, alphabetically: *HephaisTK* [10], *ICon* [9] and the post-WIMP graphical toolkit *MaggLite* [24] based on top of it, *OpenInterface* [38] (with its *OIDE* [54] and *Skemmi* [35] visual programming editors), *Squidy Lib* [31].

Data flow environments such as *EyesWeb* [25], *PureData* [45] and *Max/MSP* [7] benefit from their anteriority in comparison with these multimodal prototyping tools, as they often provide more usable visual programming development environments. Some of the authors of this report have been successfully using *PureData* as a platform for rapid prototyping of gestural interfaces [14]. A notable nice feature from these environments that could be repurposed in the ones targeted for multimodal user interfaces: the “multi-fidelity” patch/pipeline representation modes of *Cycling Max / MSP*:

1. in “edit” or “patch” mode, the dataflow representation of the pipeline, widgets of processing blocks are editable and interconnections apparent between these;
2. in “running” or “normal” mode, widgets from the pipeline are interactive, but interconnections are hidden;
3. in “presentation” mode, widgets are “ideally” positioned as it would be expected from a control GUI and connections are hidden as well.

OpenInterface/Skemmi addresses this issue with designer/developer modes and a non-linear zoom slider while *Squidy Lib* offers a zoomable user interface.

4.3.2.2. Chosen platform: *OpenInterface (OI)*

The *OpenInterface* platform [34] developed by one of the participants, Lionel Lawson, facilitates the rapid prototyping of multimodal interfaces in a visual programming environment. It also eases technically the communication between components written in different development languages (currently: C++, Java, Python,

.NET) in Windows and Linux OSes. It already features several input device components (WiiMote, webcams for computer vision, 3D mice) and some gesture recognition components, but misses a few important ones (multitouch screen/tablets, pen tablet) for the scope of our project. We decided to maintain using this platform and implement the missing components.

4.3.3. Environments for “GUI” and visualization

4.3.3.1. Existing tools

Regarding visualization, mostly libraries are available rather than rapid prototyping tools, particularly [Prefuse](#) [20] for information visualization or [VTK](#) and [Visualization Library](#) for 3D computer aided design or medical visualization. The [Processing Development Environment \(PDE\)](#) [15] simplifies the development in Java and goes further than visualization by providing other libraries for gestural input for instance. Emerging libraries such as [MT4j](#) [13] in Java and [PyMT](#) [46] in Python offer high-level “multimedia” widgets with multitouch support, yet customization of widgets still requires some effort. The more recent [VisTrails / VisMashup](#) [53] allows visual programming of workflows for data exploration and visualization.

4.3.3.2. Chosen platform: the Processing Development Environment (PDE)

We chose the Processing Development Environment (PDE) [15] since it was already mastered by the participants of the team working on designing new proposals for the graphical user interface of the annotation tool. Additionally, more scalability is offered by this solution for the prototyping: since PDE is written in Java, it is compatible with our chosen rapid prototyping platform, [OpenInterface](#) (see Section 4.3.2.2), using [proclipsing](#) [44], a bridge to the Eclipse IDE used on top of which the [OpenInterface Skemmi](#) editor is built; but it can also be re-integrated into a more standalone [MT4j](#) application if only a multitouch interface is chosen for gestural input, hence removing the dependency to [OpenInterface](#).

5. RESULTS

5.1. A tentative design towards an improved User Interface

5.1.1. Design considerations

While testing annotation tools (see Section 3), we noticed that the user experience with most of the tools was hindered due to the lack of seamless navigation techniques in lengthy signals, for instance changing the playback speed was awkward, both in terms of user input and visualization; and the related audio feedback was improperly rendered. The first task inherent to annotation is navigation into the multimedia content.

5.1.2. Mockups

We believe that a single user interface could be used for both the recording of multimodal signals and the navigation into the recorded multimedia content. Figure 4 illustrates a design proposal that would allow this combination: a standard multi-track view of audio, video, and sensor signal tracks stacked vertically is augmented with a sliding vertical zone, extending the proposal of [17] and [59], where are visualized the current frame being played in video tracks (thus behaving like a video player), and a fisheye

view of the audio waveform and sensor signals for audio and sensor tracks; the width of the zone corresponding to the same time frame for all tracks.

When recording, the zone could be located on the right, the remaining space left for visualizing past events. When navigating at a given playback speed, the zone could be located in the middle, leaving evenly proportionate space for future and past events, and restricting head movements from the user, gazing towards the center of the screen (as opposed to following visually the play head from left to right cyclicly in standard multitrack editors), the peripheral view optionally stimulated with highlighted past / future events. For a quick overview of the whole recording, the user could want to slide the zone from left to right or to a desired position as a magnifying tool.

5.1.3. Prototype

A fast prototype of the proposed design was developed using the [Processing Development Environment \(PDE\)](#) [15], as illustrated in Figure 5.



Figure 5: Screenshot of the improved user interface design proposal, prototyped into the Processing Development Environment.

5.2. Components for rapid prototyping with OpenInterface (OI)

5.2.1. Gestural input

Some device support components were previously available in the [OpenInterface](#) platform: the [Wii Remote](#) and [3D mice](#).

For the integration of multitouch devices, 2 options were available:

- capturing `WM_TOUCH` high-level events from Windows 7 using frameworks such as [MT4j](#) [13], but it requires creating applications with the chosen framework;
- accessing low-level events for devices using the [Human Interface Device \(HID\)](#) protocol (cross-platform in theory), reusing code from the [GenericHID](#) application for Windows and Linux.

We chose the second option since it also allowed with the same code base to integrate jog wheels (also using the [HID](#) protocol).

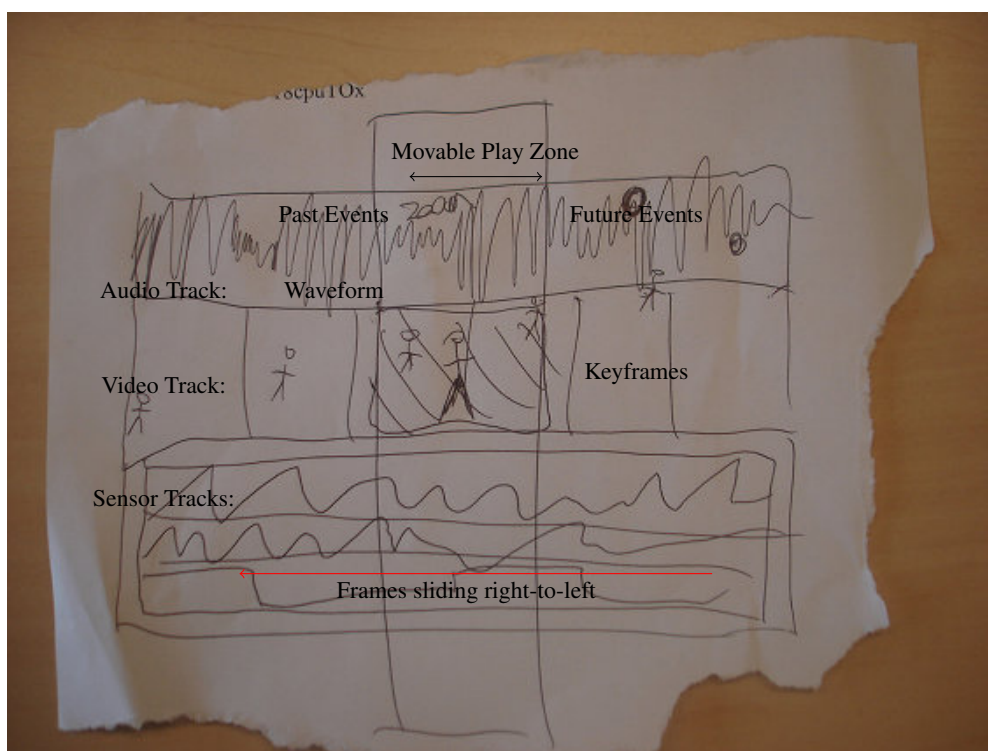


Figure 4: Annotated paper mockup of our proposed user interface design.

5.2.2. Annotation tool core/engine bindings

In this workshop we decided to adapt the already existing SSI media annotation tool to become a media annotation toolkit with a multimodal user interface. This tool consists of two parts: First, the SSI Media UI, which is a WIMP-GUI based tool to add annotations to audio and/or video data. One can operate it with mouse-clicks and a few keyboard commands. Second, the SSI core component that is responsible for the lower level signal processing. It is used by the SSI Media UI. In the course of the workshop we only adapted the SSI Media UI.

In principle, the given SSI Media toolkit was a prototypical implementation of a media annotation tool. It did not come with a special API that can be utilized from external programs. Therefore, we bundled concrete functionality of the SSI UI Media toolkit into a new interface component. The process of media annotation can be split into three subsequent process steps:

1. create and select annotation tracks (for several annotation channels)
2. segment and reorder segments in one annotation track (includes selection of segments)
3. edit (annotate) segment meta data

This process steps can be performed by the extracted functionality that includes among others start/stop playing of annotation segments, edit of annotations, selection of next/ previous segments, etc. We needed now a way to plug other input modalities to the tool that use this extracted functionality.

We created an OpenInterface component for the adapted SSI Media UI. This OI component allows to use the provided SSI Me-

dia functionality by other OI input components. Figure 6 shows several improvements we applied to its GUI.

5.3. Testing multimodal pipelines with OpenInterface (OI)

First, we connected the following interaction devices in a new OI project to the SSI OI component. This included a WWI-mote, a 3Dconnexion SpaceNavigator mouse and a Contour Design Xpress jogwheel. Moreover, we created a new speech input OI component for the Julius toolkit [julius.sourceforge.jp/en]. Additionally, we integrated mouse behavior not only by clicking but also with mouse gestures. Each modality (a modality is an interaction device with a dedicated interaction language) was then coupled with specific functionality of the SSI Media toolkit. Not all modalities fit well for the all of the functionality, but this relies to higher-level interaction design. For example, it might be a good idea to select annotation tracks via speech input (command: "next track"). Within such a track one uses mouse gestures to select next and previous segments. When a distinct segment is selected, one uses speech input again to start and stop playing the media (e.g., command: "play segment"). And the practicability of the proposed modalities varies. A wii-mote that fits well for arm-based gestures is not the first choice for smaller gestures for media annotation. On the other hand, a jogwheel was invented to improve video editing, thus fitting better for our work. Future work will include research on an improved interaction design, utilizing the "right" modalities for media annotation with the SSI_UI.



Figure 6: Screenshot of the improved GUI of the SSI annotation tool, integrated into the OpenInterface platform as component.

6. FUTURE WORK

6.1. Integration into MediaCycle, a framework for multimedia content navigation by similarity

MediaCycle is a framework for multimedia content browsing by similarity, developed within the numediart Research Program in Digital Art Technologies, providing componentized algorithms for feature extraction, classification and visualization. The supported media types are: audio (from loops to laughter) [11], video (particularly featuring dancers) [56], images...

This framework already solves some of the issues raised in Section 5.1.1 by providing flexible audiovisual engines for the navigation in multimedia content (audio feedback with variable playback speed and visual feedback with cost-effective zoom and animated transitions). Moreover, the interoperation of an annotation timeline (displaying a few elements of the recorded database) with a browser view (displaying the whole database at different levels of segmentation) such as the one already provided by MediaCycle could help compare annotations between recordings and segments. Finally, the use of this framework could help reduce the number of video keyframes by content-based grouping of frames, with a possible scalability against the user-defined zoom factor.

6.2. Usability testing

We received some feedback from several eINTERFACE participants who had already had to use an annotation tool, regarding their satisfaction with the tool they used. After the setup of a detailed protocol, usability tests based on simple tasks will be per-

formed with the prototype, trying to determine if the user interface improves the annotation efficiency and pleurability.

7. CONCLUSIONS

We reached a first step towards more usable annotation tools for multimedia content: we raised the problems with current tools and proposed a new design to overcome these issues. The prototype needs to be polished and tested with users to validate the design.

Meaning to produce deliverables available to most people (low-cost, open-source, and so on...) the eINTERFACE way, we developed:

- a free and opensource toolbox, mostly based on cross-platform tools and libraries;
- compatibility with low-cost input devices;
- a starting point to undertake usability testing that demonstrate the validity of the proposed solution.

8. ACKNOWLEDGEMENTS

Christian Frisson works for the numediart long-term research program centered on Digital Media Arts, funded by Région Wallonne, Belgium (grant N°716631).

Ceren Kayalar's PhD Research Project is partly funded by TUBITAK Career Research Grant 105E087 of her advisor, Dr. Selim Balcisoy.

Florian Lingenfeller and Johannes Wagner are funded by the EU in the CALLAS Integrated Project (IST-34800).

We would like to thank the eINTERFACE participants who provided us some feedback on their use of annotation tools and remarks on our design, notably Ismail Ari, Dennis Reidsma [47] and Albert Ali Salah.

We would like to thank all members of the eINTERFACE'10 organizing committee for ensuring a tight workflow (and social events) throughout the workshop.

9. REFERENCES

9.1. Scientific references (books, journals, conferences, workshops)

- [2] Maneesh Agrawala and Michael Shilman. "DIZI: A Digital Ink Zooming Interface for Document Annotation". In: *Proceedings of INTERACT*. 2005. URL: <http://graphics.stanford.edu/papers/dizi/DIZI.3.pdf>. P.: 49.
- [3] Wolfgang Aigner et al. "Visual Methods for Analyzing Time-Oriented Data". In: *IEEE Transactions on Visualization and Computer Graphics* 14.1 (2008). Pp. 47–60. URL: <http://www.informatik.uni-rostock.de/~ct/Publications/tvcg08.pdf>. P.: 48.
- [5] Anastasia Bezerianos, Pierre Dragicevic, and Ravin Balakrishnan. "Mnemonic Rendering: An Image-Based Approach for Exposing Hidden Changes in Dynamic Displays". In: *Proceedings of UIST 2006 - ACM Symposium on User Interface Software and Technology*. 2006. Pp. 159–168. URL: <http://www.dgp.toronto.edu/~anab/mnemonic/>. P.: 48.
- [6] Tony Bigbee, Dan Loehr, and Lisa Harper. *Emerging Requirements for Multi-Modal Annotation and Analysis Tools*. Tech. rep. The MITRE Corporation, 2001. URL: http://www.mitre.org/work/tech_papers/tech_papers_01/bigbee_emerging/bigbee_emerging.pdf. P.: 46.
- [8] Stefanie Dipper, Michael Götze, and Manfred Stede. "Simple Annotation Tools for Complex Annotation Tasks: an Evaluation". In: *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*. 2004. Pp. 54–62. URL: <http://www.ling.uni-potsdam.de/~7Edipper/papers/xbrac04-sfb.pdf>. P.: 46.
- [11] Stéphane Dupont et al. "Browsing Sound and Music Libraries by Similarity". In: *128th Audio Engineering Society (AES) Convention*. 2010. P.: 52.
- [12] L. Dybkjaer and N. O. Bernsen. "Towards general-purpose annotation tools: how far are we today?" In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation LREC'2004*. 2004. URL: <http://www.nis.sdu.dk/~nob/publications/LREC2004-annotation-DybkjaerBernsen.pdf>. Pp.: 46–48.
- [14] Christian Frisson et al. "DeviceCycle: rapid and reusable prototyping of gestural interfaces, applied to audio browsing by similarity". In: *Proceedings of the New Interfaces for Musical Expression++ (NIME++)*. 2010. ISBN: 978-0-646-53482-4. URL: http://www.educ.dab.uts.edu.au/nime/PROCEEDINGS/papers/Demo%20Q1-Q15/P473_Frisson.pdf. P.: 49.
- [16] Patrick Gebhard et al. "Authoring Scenes for Adaptive, Interactive Performances". In: *Proceedings of the ACM AAMAS*. 2003. P.: 46.
- [17] Kristian Gohlke et al. "Track Displays in DAW Software: Beyond Waveform Views". In: *Audio Engineering Society Convention 128*. 2010. Pp.: 48, 50.
- [18] Joey Hagedorn, Joshua Hailpern, and Karrie G. Karahalios. "VCode and VData: Illustrating a new Framework for Supporting the Video Annotation Workflow". In: *Proceedings of AVI*. 2008. Pp.: 47, 49.
- [19] Björn Hartmann et al. "Authoring Sensor-based Interactions by Demonstration with Direct Manipulation and Pattern Recognition". In: *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI)*. 2007. P.: 46.
- [20] Jeffrey Heer, Stuart K. Card, and James A. Landay. "Prefuse: A Toolkit for Interactive Information Visualization". In: *ACM Human Factors in Computing Systems (CHI)*. 2005. URL: <http://vis.berkeley.edu/papers/prefuse/>. P.: 50.
- [21] Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. "Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations". In: *Proceedings of CHI*. 2009. P.: 48.
- [22] Jeffrey Heer and George Robertson. "Animated Transitions in Statistical Data Graphics". In: *IEEE Information Visualization (InfoVis)*. 2007. URL: http://vis.berkeley.edu/papers/animated_transitions/. P.: 48.
- [23] A. Heloir, M. Neff, and M. Kipp. "Exploiting Motion Capture for Virtual Human Animation". In: *Proceedings of the Workshop "Multimodal Corpora: Advances in Capturing, Coding and Analyzing Multimodality" at LREC-2010*. 2010. URL: <http://embots.dfki.de/doc/HeloiRetal10.pdf>. P.: 46.
- [26] Alexander Refsum Jensenius. "Using Motiongrams in the Study of Musical Gestures". In: *ICMC 2006*. 2006. URL: <http://www.hf.uio.no/imv/forskning/forskningsprosjekter/musicalgestures/publications/pdf/jensenius-icmc2006.pdf>. P.: 48.
- [27] Ceren Kayalar, Emrah Kavlak, and Selim Balcisoy. "A User Interface Prototype For A Mobile Augmented Reality Tool To Assist Archaeological Fieldwork". In: *SIGGRAPH'08: ACM SIGGRAPH 2008 Posters*. 2008. URL: http://students.sabanciuniv.edu/~ckayalar/siggraph08_poster_kayalar_kavlak_balcisoy.jpg. P.: 46.
- [29] Michael Kipp. "Spatiotemporal Coding in ANVIL". In: *Proceedings of the 6th international conference on Language Resources and Evaluation (LREC-08)*. 2008. URL: http://embots.dfki.de/doc/Kipp08_Anvil.pdf. Pp.: 46, 47.
- [30] Nicholas Kong and Maneesh Agrawala. "Perceptual Interpretation of Ink Annotations on Line Charts". In: *Proceedings of UIST*. 2009. P.: 49.

- [32] Johannes Kopf et al. “Capturing and viewing gigapixel images”. In: *SIGGRAPH’07: ACM SIGGRAPH 2007 papers*. San Diego, California 2007. P.: 46.
- [33] Rony Kubat et al. “TotalRecall: Visualization and Semi-Automatic Annotation of Very Large Audio-Visual Corpora”. In: *Ninth International Conference on Multimodal Interfaces (ICMI 2007)*. 2007. URL: http://www.media.mit.edu/cogmac/publications/kubat_icmi2007.pdf. P.: 48.
- [34] Jean-Yves Lionel Lawson et al. “An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components”. In: *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems (EICS’09)*. 2009. P.: 49.
- [36] Hyowon Lee et al. “Implementation and analysis of several keyframe-based browsing interfaces to digital video”. In: *in Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*. Springer, 2000. Pp. 206–218. P.: 49.
- [37] Michael Lew. “Live Cinema: Designing an Instrument for Cinema Editing as a Live Performance”. In: *Proceedings of New Interfaces for Musical Expression (NIME)*. 2004. URL: http://nime.org/2004/NIME04/paper/NIME04_3A03.pdf. P.: 46.
- [39] Bill Moggridge. *Designing Interactions*. The MIT Press, 2007. ISBN: 9780262134743. URL: <http://www.designinginteractions.com>. P.: 49.
- [40] Michael Nunes et al. “What Did I Miss? Visualizing the Past through Video Traces”. In: *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW’07)*. 2007. URL: <http://grouplab.cpsc.ucalgary.ca/grouplab/uploads/Publications/Publications/2007-VideoTraces.ECSCW.pdf>. Pp.: 48, 49.
- [42] Andrei Popescu-Belis. “Multimodal Signal Processing: Theory and applications for human-computer interactions”. In: ed. by Jean-Philippe Thiran, Ferran Marqués, and Hervé Boulard. Elsevier, 2009. Chap. Managing Multimodal Data, Metadata and Annotations: Challenges and Solutions, pp. 207–228. ISBN: 978-0-12-374825-6. P.: 46.
- [43] Yannick Prié, Olivier Aubert, and Bertrand Richard. “Démonstration: Advene, un outil pour la lecture active audiovisuelle”. In: *IHM’2008*. 2008. URL: <http://liris.cnrs.fr/advene/doc/advene-demo-ihm08.pdf>. P.: 47.
- [47] Dennis Reidsma. “Annotations and Subjective Machines — of annotators, embodied agents, users, and other humans”. PhD thesis. University of Twente, 2008. DOI: [10.3990/1.9789036527262](https://doi.org/10.3990/1.9789036527262). P.: 53.
- [48] Dennis Reidsma, Dennis H. W. Hof, and Natav sa Jovanović. “Designing Focused and Efficient Annotation Tools”. In: *Measuring Behaviour*. Ed. by L. P. J. J. Noldus et al. Wageningen, NL 2005. Pp. 149–152. ISBN: 90-74821-71-5. URL: <http://doc.utwente.nl/65561/1/reidsmaMB05.pdf>. P.: 46.
- [49] Richard Rinehart. “The Media Art Notation System: Documenting and Preserving Digital/Media Art”. In: *Leonardo* 40.2 (Apr. 2007). 2. Pp. 181–187. P.: 46.
- [50] Katharina Rohlfing et al. *Comparison of multimodal annotation tools*. Tech. rep. Gesprächsforschung - Online-Zeitschrift zur verbalen Interaktion, 2006. URL: <http://www.gespraechsforschung-ozs.de/heft2006/tb-rohlfing.pdf>. P.: 46.
- [51] Natalie Ruiz, Fang Chen, and Sharon Oviatt. “Multimodal Signal Processing: Theory and applications for human-computer interaction”. In: ed. by Jean-Philippe Thiran, Ferran Marqués, and Hervé Boulard. Elsevier, 2009. Chap. Multimodal Input, pp. 231–256. ISBN: 978-0-12-374825-6. P.: 48.
- [52] Dan Saffer. *Designing Gestural Interfaces*. O’Reilly Media, Inc., 2009. ISBN: 978-0-596-51839-4. URL: <http://www.designinggesturalinterfaces.com/>. P.: 49.
- [53] Emanuele Santos et al. “VisMashup: Streamlining the Creation of Custom Visualization Applications”. In: *IEEE Visualization*. 2009. URL: <http://www.cs.utah.edu/~juliana/pub/mashup-vis2009.pdf>. P.: 50.
- [55] Noah Snaveley, Steven M. Seitz, and Richard Szeliski. “Photo tourism: exploring photo collections in 3D”. In: *SIGGRAPH’06: ACM SIGGRAPH 2006 Papers*. Boston, Massachusetts 2006. ISBN: 1-59593-364-6. DOI: <http://doi.acm.org/10.1145/1179352.1141964>. P.: 46.
- [56] Damien Tardieu et al. “An interactive installation for browsing a dance video database.” In: *IEEE International Conference on Multimedia & Expo*. 2010. Pp.: 46, 52.
- [61] Johannes Wagner, Elisabeth André, and Frank Jung. “Smart sensor integration: A framework for multimodal emotion recognition in real-time”. In: *Affective Computing and Intelligent Interaction (ACII 2009)*. 2009. URL: http://mm-werkstatt.informatik.uni-augsburg.de/files/publications/261/ssi_acii09_camera.pdf. Pp.: 47, 49.
- [62] Johannes Wagner et al. “SSI/ModelUI - A Tool for the Acquisition and Annotation of Human Generated Signals”. In: *DAGA*. 2010. URL: http://mm-werkstatt.informatik.uni-augsburg.de/files/publications/295/wagner_daga2010.pdf. Pp.: 47, 48.
- [63] Colin Ware. *Visual Thinking: for Design*. Interactive Technologies. Morgan Kaufmann, 2008. ISBN: 978-0123708960. P.: 48.

9.2. Software (annotation tools, rapid prototyping frameworks...)

- [1] “Advene (Annotate Digital Video, Exchange on the Net)”. URL: <http://www.advene.org>. P.: 47.
- [4] “AmiGram: AMI Graphical Representation and Annotation Module”. URL: <http://ami.dfki.de/amigram/>. P.: 47.
- [7] Cycling’74. “Max/MSP”. URL: <http://www.cycling74.com>. P.: 49.
- [9] Pierre Dragicevic, Jean-Daniel Fekete, and Stéphane Huot. “Icon (Input Configurator)”. URL: <http://inputconf.sourceforge.net>. P.: 49.

- [10] Bruno Dumas. “HephaisTK”. URL: <http://sourceforge.net/projects/hephaistk/>. P.: 49.
- [13] Fraunhofer - Institute for Industrial Engineering. “MT4j - Multitouch for Java™”. URL: <http://www.mt4j.org>. P.: 50.
- [15] Ben Fry and Casey Reas. “Processing Development Environment (PDE)”. URL: <http://www.processing.org>. P.: 50.
- [24] Stéphane Huot and Cédric Dumas. “MaggLite”. URL: <http://www.emn.fr/x-info/magglite/>. P.: 49.
- [25] DIST-University of Genova InfoMus Lab. “The EyesWeb XMI (eXtended Multimodal Interaction) platform”. Version 5.0.2.0. URL: <http://www.eyesweb.org>. P.: 49.
- [28] Michael Kipp. “ANVIL: The Video Annotation Research Tool”. URL: <http://www.anvil-software.de>. P.: 47.
- [31] Werner A. König, Roman Rädle, and Harald Reiterer. “Squidy Lib”. URL: <http://www.squidy-lib.de>. P.: 49.
- [35] Lionel Lawson and Amro Al-Akkad. “Skemmi, an Eclipse based front-end to OpenInterface Runtime”. URL: <https://forge.openinterface.org/projects/skemmi/>. P.: 49.
- [38] Lionel Lawson et al. “The OpenInterface platform”. URL: <http://www.openinterface.org>. P.: 49.
- [41] IRI / Centre Pompidou. “Lignes de Temps”. URL: <http://www.iri.centrepompidou.fr>. P.: 47.
- [44] “proclipsing - Eclipse Processing Development Tools”. URL: <http://code.google.com/p/proclipsing/>. P.: 50.
- [45] Miller Puckette and all PureData developers. “PureData”. URL: <http://www.puredata.info>. P.: 49.
- [46] “PyMT”. URL: <http://pymt.txzone.net>. P.: 50.
- [54] Marcos Serrano and Michael Ortega. “The OI Interaction Development Environment (OIDE)”. URL: <https://forge.openinterface.org/projects/oide/>. P.: 49.
- [57] The Technical Group of the Max Planck Institute for Psycholinguistics. “ELAN”. URL: <http://www.la-mpi.eu/tools/elan>. P.: 47.
- [58] “VCode & VData: Video Annotation Tools”. URL: <http://social.cs.uiuc.edu/projects/vcode.html>. P.: 47.
- [59] VeriCorder. “1st Video: Mobile Video Editing Software”. URL: <http://www.vericorder.com>. P.: 50.
- [60] Johannes Wagner. “Smart Sensor Integration (SSI)”. URL: <http://mm-werkstatt.informatik.uni-augsburg.de/ssi.html>. P.: 47.
- [64] David Young. “On The Mark”. URL: <http://onthemark.sourceforge.net>. P.: 47.