UNIVERSITÉ CÔTE D'AZUR
SI5/M2 INFO
2024-2025

Julien Deantoni

# Domain Specific Language

## VideoML

## Project delivery

This project concerns the second delivery of the DSL course. It will include the implementation of a DSL focused on the specification of video edition, in either an internal or an external DSL. Deliveries are expected by email (to Julien Deantoni: firstname.lastname@univ-cotedazur.fr, with [DSL] as object prefix) followed by "team X lab 2" where X is the name of your team. The delivery is expected before the $19^{th}$ of January 2025 at 10:00PM Paris Time. The delivery is expected as a PDF report (please let your report be succinct and rigorous).
The report must contain :

- the name of the members of your team
- a link to the code of your DSL
- a description of the language proposed:
  - the domain model represented as a class diagram;
  - the concrete syntax represented in a BNF-like form;
  - a description of your language and how it was implemented;
  - a simple description on how you wrote the compiler to obtain executable code
- a set of relevant scenarios implemented by using your language(s) (internal or external);
- a critical analysis of (i) DSL implementation with respect to the VideoML use cases and (ii) the technology you chose to achieve it;
- responsibility of each member in the team with respect to the delivered project.
- a small video example whose editing is realized with your DSL to demonstrate its usage and all of its features. The associated script and materials will be provided.

## Objectives: Define the VideoML language

Your objective here is to define a DSL allowing the definition of various, possibly complex, video editing. Classic functionalities contain video loading, cutting, concatenations and title insertions. But it can bring more functionalities like for instance special effects in transitions (e.g., fade in) or clip resolution alteration or clip rotation. The format of supported video files is not important as long as it is not a confidential one.

From your programs written in the $VideoML$ language, you will generate executable code. The target language is your choice, but it should be usable under different operating systems. There exist many libraries for video edition like for instance (not necessarily the most interesting ones) Xuggle for java (`https://github.com/artclarke/humble-video`), MoviePy for python (`https://github.com/Zulko/moviepy`) or openShot for C++ (`https://www.openshot.org/libopenshot/`). A more complete list is available at `https://github.com/topics/video-editing?o=desc&s=stars` and many more exist.

## Basic scenarios

1. **Scenario 1**

   Bob and Alice went on holiday together, and they did some videos. Now they want to send a consolidated video to their family. Consequently, they want to:

   (a) add a title on a black background at the beginning for 10 seconds stating where they were and when.

   (b) add a first video clip that appears directly after the title screen

   (c) add another video clip that appears directly after the first clip

   (d) add a thanks sentence at the end, lasting for 15 seconds

   (e) export the result as a video file

2. **Scenario 2**

   Alice needs to do a cast for her work. She needs to:

   (a) add an introduction title on a black background at the beginning for 10 seconds.

   (b) load a first video clip $clip1$

   (c) create two clips $clip1a$ and $clip1b$ respectively taken from 00:23 to 01:47 min and from 02:01 to 02:21 min.

   (d) add $clip1a$ with a subtitle $s1$ from the beginning and for 10 seconds, followed by another subtitle starting 30 seconds after the end of the first one and visible for 10 seconds.

   (e) add $clip1b$ with a subtitle starting 5 seconds before $clip1b$ and lasting for 15 seconds.

   (f) add a thanks conclusive text on a fixed background color (let's say black).

   (g) export the result as a video file

## Commmon Parts

The following parts must be available in your DSL:

- **Domain Model**:

  The domain model (a.k.a. abstract syntax) should be clearly identified in the delivered code. It will be provided as a class diagram, together with explanation about the main choices you did.

- **Concrete syntax**:

  The concrete syntax (external or embedded), in a BNF form, must be clearly identified and used by a relevant set of scenarios. The syntax must leverage the tool chosen to implement it to make it clear and easy to use.

- **Validation**:

  Support your end-user by checking that a model is realizable on the expected platform. For example, choice of video format, length and timings are consistent , etc.

- **Code generation**:

  Provide a generator producing code whose execution results in one or more playable video(s). This code can be directly compiled by your script generation (if needed), and played to simplify the user task. The target language is your choice.

- **DSL Addons for User Acceptance** The acceptance of a language by users is usually not only due to the language itself but also on the extra services provided by the languages. These services can be related to the language itself (like the borrowchecker in Rust) or used here to help the user in the definition of its programs (like for instance a graphical representation of the control flow or a very precise and didactic error spotting). You will provide at least one extra service for the VideoML DSL.

## “À la carte” features

The remainder of this document describes some extensions that you can implement on top of your language. Each extension is defined by a short description and at least one acceptance scenario.
Choose at least one (preferably two) feature(s) to introduce in your project. Feel free to propose other features you would like to develop.

### Support for stacking and transitions between clips

This extension makes it possible to add transitions between the various concatenated clips. It also enables stacking multiple clips on top of each other with specific sizes.
scenario: Bob wants to show himself commenting a specific sport sequence. So he wants to add a video of himself commenting a scene in a corner of the main video. Additionally, he wants to add fade out fade in transitions when creating the scene to be commented. Of course the fade effect does not affect the video of himself, which stay in the corner.

### Support for tuned animated texts

To make generated videos more appealing, it is interesting to have some animated text, either to put text scrolling at the beginning or end of the video or to emphasize some part of a text.
scenario1: Jackie wants to add the full list of contributors at the end of her video. She would like scrolling text for that. Also, she wants to be able to define the speed of the scrolling text.
scenario2: Jackie wants to apply effects to the text at the start of her presentation, as it showcases the name of her tool. The aim is to capture the audience’s attention.

### Support for audio

Some people like to add some audio to their videos. This extension is there to help them. Several audio tracks should be supported with the possibility to cut and concatenate them. Having transition effects is a plus.
scenario: Andreï wants to add oral explanation to a video (possibly made up with the concatenation of various small clips). For that he recorded several small explanations, and he wants to synchronize them with the video. He would also like to be able to adjust the volume of each audio chunk, but this is not mandatory.

### Support video clip tuning and effects

It may be interesting to adjust the rendering parameters of each clip since they may have been recorded in different situations. For instance brightness or saturation may be adjusted. More interestingly, it should be possible to crop a part of a video or to add specific effects like for instance fixing the frame at time 00:43 for 12 seconds or zooming a specific part of the video. Any other tuning or effect are encouraged.

scenario: Nawa wants to concatenate videos created on different operating systems for the same software. To ensure consistency, she needs the ability to crop OS-specific elements, such as taskbars and window decorations. Additionally, she wants to freeze frames at specific times and for specific duration, allowing viewers enough time to read the text on the screen. Finally, she plans to convert the entire video to grayscale.