1 Covida

Annotating videos is an exhausting task, but it has a great benefit for several communities. Thus labeled ground truth data is the foundation for supervised machine learning approaches. Covida offers an easy accessible interface for annotating video data in single user enviroment, for example Asus EEE Slate, and multi user enviroment, like collaborative work on a pen enabled multi touch table.

Figure 1: Covida screenshot.

Figure 1 introduces the interface of the key features provided by Covida. Covida offers the possibility to annotate videos directly via touch gestures, drawings and handwriting. For outlining and labeling the shape of an object, Covida provides a pen-based interface which combines pen and touch input. User studies showed that especially for complex structures the usage of a pen device improves the effectiveness of the outlining process. Annotations can be searched via handwritings on the search field (See Figure 1 upper left corner) which can easily be accessed via touch gestures. Further annotation text can be stored in the clipboard (See Figure 1 bottom left corner) via dragging of the annotation text via touch drag gestures. In a similar way annotations can be assigned to semantic groups like a name of a person to the semantic group 'person'. Further the stored annotation data is saved as aRDF data in a XML file to grant the possibilty to modify and visualize the labled annotation data outside of the Covida application.

2 Architecture

The CoVidA application architecture is structured as shown in Figure 2. All core functionallities are bundled in the covida-core package. All visual components are handled from the visual package, which is exchangeable. In this setup the covida-visual-jme2 package is used to render the covida application. The video package is also exchangeable and currently is the covida-video-vlcj package used which has the vlcj as foundation.
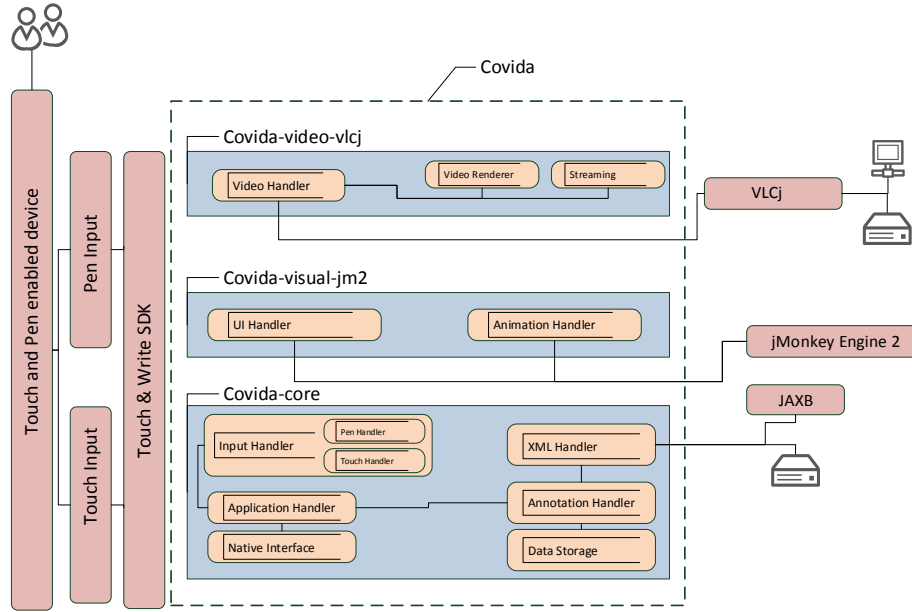

Figure 2: Covida Architecture.

In the current setup (Figure 2) three third party components are used. First the TouchAndWrite SDK which is used as basis for the covidacore component. Second the VLCj component which provides the videovlcj component whith methods to handle a huge variaty of video formats. Finally the jMonkey Engine 2 (jme2) is a scene graph based 3D engine framework completly written in Java2. The CoVidA presentation layer gives with aid of the jMonkey Engine 2 framwork graphical feedback onto user interactions. With this framework CoVidA has the possibility to move and transform multiple videos and switch textures directly. For other feedback, like opening or closing, the jMonkey Engine 2 offers CoVidA the possibility to animate the user interface objects and control this animations in case they must be interupted.

3 Implementation

4 Conclusion and Future Work