Journal of
open research software

Pastell, M 2016 CowLog – Cross-Platform Application for Coding Behaviours from Video. *Journal of Open Research Software*, 4: e15, DOI: http://dx.doi.org/10.5334/jors.113

SOFTWARE METAPAPER

CowLog – Cross-Platform Application for Coding Behaviours from Video

Matti Pastell[1]

[1] Natural Resources Institute Finland (Luke), Green Technology. Viikinkaari 4, 00790 Helsinki, Finland
matti.pastell@luke.fi

CowLog is a cross-platform application to code behaviours from video recordings for use in behavioural research. The software has been used in several studies e.g. to study sleep in dairy calves, emotions in goats and the mind wandering related to computer use during lectures.

CowLog 3 is implemented using JavaScript and HTML using the Electron framework. The framework allows the development of packaged cross-platform applications using features from web browser (Chromium) as well as server side JavaScript from Node.js. The program supports using multiple videos simultaneously and HTML5 and VLC video players.

CowLog can be used for any project that requires coding the time of events from digital video. It is released under GNU GPL v2 making it possible for users to modify the application for their own needs. The software is available through its website http://cowlog.org.

Keywords: Behavioural coding; video coding; Electron Framework

(1) Overview

Introduction

CowLog is a cross-platform application to code behaviours from video recordings. It can be used in any application where coding behaviours from video is needed. The first version of the application [1] of was developed in 2009 for a project where we needed to code the sleep behaviour of calves from video recordings in order to develop a sleep actigraph [2]. The software has also been used in several other studies e.g. to study emotions in goats [3], the mind wandering related to computer use during lectures [4] and effects of simulated transport on the behaviour of eastern blue tongued lizards [5]. There were 33 papers indexed in Scopus citing the first paper about CowLog [1] at the time of the writing.

The initial release run only on Linux and had limited configuration options [1]. It soon became clear from our own projects and user feedback from other groups after the initial release that there was a clear need for a cross-platform version.

Two versions of CowLog were developed to address these issues: Python based desktop version and HTML5 application that runs in the browser. Several users were concerned about the privacy issues of the web application although it only used single script to format the output file and everything else run on client side.

It was decided to direct development effort only to JavaScript and in 2015 the web version of the application was ported to the Electron Framework and packaged beta version as a desktop application. This was released as CowLog 3 in early 2016.

The aim of this paper is to describe the implementation of CowLog 3 in order to make it easier for users to modify the program for their needs or participate in development of the released version.

Implementation and architecture

CowLog version 3 described in this paper was implemented using JavaScript and HTML5 using the Electron framework. The framework allows the development of packaged cross-platform applications using features from the Chromium web browser as well as server side JavaScript from Node.js. This version is based on earlier version of CowLog that runs in the web browser.

An application implemented using the Electron framework has a main process and one or several renderer processes which display the user interface. The main process, based on Node.js, is used to create the user interface windows, controls the application lifetime and provides features such as file system access and via Node.js modules. The renderer process displays the user interface as a website, each created window runs in its own process.

The renderer process and main process communicate with event based system using the Electron interprocess communication module. An application using Electron can also use almost any Node.js module to extend the functionality.

The used architecture makes the application very modular and the design and implementation in one window can be changed easily without affecting the other windows as long as the messages exchanged between processes are not changed.

The user interface makes extensive use of the jQuery JavaScript library. For instance the coding buttons in the user interface are generated dynamically based on the project configuration and the program indicates the currently active code for each behavioural class. The jQuery UI library is used for the video progress bar and date picker components. The user interface uses the Bootstrap framework for layout and design of the elements such as buttons and forms.

The ethogram, folders for videos and output, used video player, modifiers for behaviours and keyboard shortcuts for coding can be defined and modified using CowLog projects. The project settings are saved to JSON file which can be used to transfer the settings between computers and e.g. shared together with publications.

The software makes use of two video players and the user can select the most suitable one for their application:

1. Native HTML5 video player is able to play ogg, mp4 (h.264) and webm video formats. The JavaScript API of the player is easy to use and has good documentation.
2. VLC player controlled using WebChimera.js, which is able to render video frames directly on HTML5 canvas giving good performance. CowLog binaries on Windows and Mac OS X are bundled with VLC player and precompiled WebChimera.js. On Linux the user should install VLC player separately.

The program supports using multiple videos simultaneously, and all of the videos are played in single window. The playback speed of the video can be changed to make it slower or faster. There is no limit for the number of simultaneous videos used, the practical limitations depends on the capacity of the used computer. In our practical projects we have used at most 3 cameras at the same time to cover large enough area. The layout of the videos is also user selectable.

CowLog user interface for the actual behavioural coding is focused around buttons representing each behaviour. The user can also configure and use keyboard shortcuts for the coding.

Dialogs are used to create and edit projects and starting or resuming coding sessions. Coding buttons and the videos are displayed in separate windows to make it easier to use the software with multiple displays. **Figure 1** shows the CowLog main window configured with a simple ethogram to study the behaviour of sows [6] around farrowing and the video window.

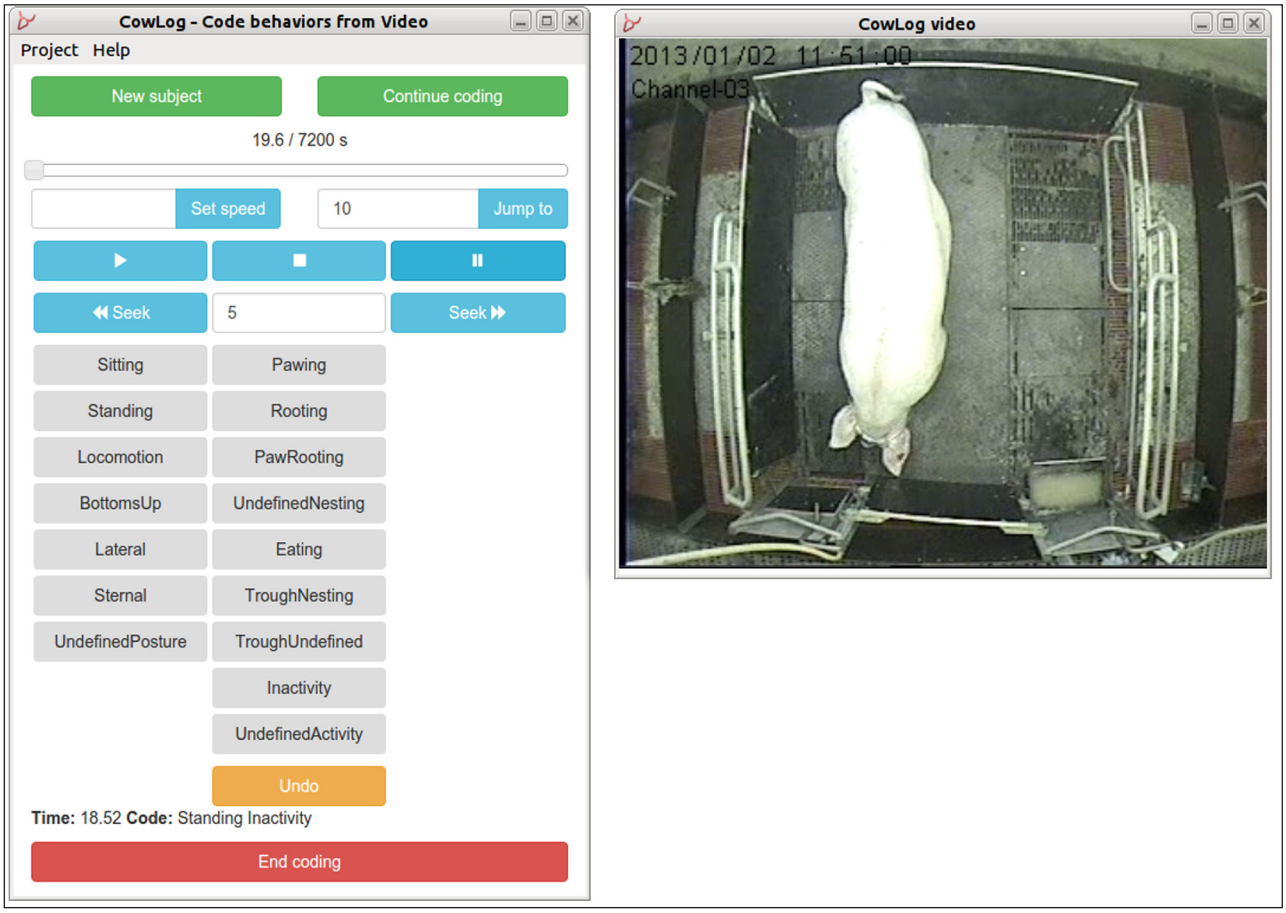

Figure 1: CowLog coding window (left) configured to study the behaviour of pigs before farrowing and the video window (right). The coding window shows the last entered code and allows the control of video.

Releases of the software are packaged for different platforms using electron-packager, which makes it simple to package the release for Mac, Windows and Linux architectures with a single command. The Windows version is also released with an installer built using Inno Setup.

Quality control

The software goes trough functional testing and a beta release before each final release using existing project configurations and several video formats. There are currently no automated tests. The development takes place on Ubuntu Linux, Mac OS X and Windows and the builds are tested on each of the platforms.

The software is used in our own research and we get immediate feedback from users. Several other research groups also use CowLog and have reported from their experiences.

All users can report bugs and ask for features on GitHub or using e-mail.

(2) Availability

Operating system

Windows, 7, 8 and 10. Mac OS X (> = 10.8). Linux 32bit and 64bit, tested on Ubuntu 14.04 and 15.10. Electron runs also on Linux ARM v7, but CowLog has not been tested on that platform.

Programming language

JavaScript and HTML5 running on the Electron framework.

Additional system requirements

A typical coding session has modest system requirements. The system requirements depend on the codec, resolution and the number of simultaneous videos used. The program can run on all Electron supported platforms.

Dependencies

- Electron 0.36.2
- jQuery 2.1.4
- jQuery UI 1.8.21
- Bootstrap 3.3.5
- WebChimera.js 0.1.47
- VLC player 2.2 (Optional)

List of contributors

- Dr. Matti Pastell (Natural Resources Institute Finland, Luke). Developer.
- Dr. Laura Hänninen (University of Helsinki). Requirements for the first version
- Dr. Andrew Robins (The University of Queensland). CowLog logo, testing and feedback.

Software location

Archive

Name: GitHub

Persistent identifier: https://github.com/mpastell/CowLog/releases

Licence: GPL-v2.0

Publisher: Matti Pastell

Date published: 13/01/2016 (v 3.0.1)

Code repository

Name: GitHub

Identifier: https://github.com/mpastell/CowLog

Licence: GPL-v2.0

Date published: 12/09/2015

Language

English

(3) Reuse potential

CowLog is general purpose application in behavioural coding. It can be used for any project that requires coding the time of events from digital video. The software has been used in studies about animal behaviour as well facial expressions in humans. The software could easily be modified to do also live coding. New developers are welcome to make contribution to the program via GitHub.

Competing Interests

The author declares that they have no competing interests.

Acknowledgements

I acknowledge all CowLog users who have sent me their feedback and used the software in interesting research projects.

References

1. **Hänninen, L** and **Pastell, M** 2009 CowLog: Open source software for coding behaviors from digital video. *Behavior Research Methods,* 41 (2): 472–476. DOI: http://dx.doi.org/10.3758/BRM.41.2.472
2. **Hokkanen, A-H, Hänninen, L, Tiusanen, J** and **Pastell, M** 2011 Predicting sleep and lying time of calves with a support vector machine classifier using accelerometer data. *Applied Animal Behaviour Science,* 134(1–2): 10–15, ISSN 0168-1591. DOI: http://dx.doi.org/10.1016/j.applanim.2011.06.016
3. **Briefer, E F, Tettamanti, F** and **McElligott, A G** 2015 Emotions in goats: Mapping physiological, behavioural and vocal profiles. *Animal Behaviour,* 99: 131–143. DOI: http://dx.doi.org/10.1016/j.anbehav.2014.11.002
4. **Risko, E F, Buchanan, D, Medimorec, S** and **Kingstone, A** 2013 Everyday attention: Mind wandering and computer use during lectures. *Computers and Education,* 68: 275–283. http://dx.doi.org/DOI: 10.1016/j.compedu.2013.05.001
5. **Mancera, K, Murray, P J, Gao, Y N, Lisle, A** and **Phillips, C** 2014 The effects of simulated transport on the behaviour of eastern blue tongued lizards (Tiliqua scincoides). *Animal Welfare,* 23 (3): 239–249. DOI: http://dx.doi.org/10.7120/09627286.23.3.239
6. **Pastell, M, Hietaoja, J, Tiusanen, J, Yun, J** and **Valros, A** 2014 A Model to Detect Farrowing based on Sow Activity. *Proceedings of the International Conference of Agricultural Engineering - AgEng 2014,* Zurich. DOI: http://dx.doi.org/10.13140/RG.2.1.2073.0960

How to cite this article: Pastell, M 2016 CowLog – Cross-Platform Application for Coding Behaviours from Video. *Journal of Open Research Software,* 4: e15, DOI: http://dx.doi.org/10.5334/jors.113

Submitted: 14 January 2016 **Aceepted:** 13 April 2016 **Published:** 27 April 2016

]u[*Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS