Software as academic output

Caroline Jay and Robert Haines
Software is now considered an academic output

- When does software ‘count’ as an output?
Historically (currently)

• Software is hidden
  – Describe what it does
  – Describing how it does it (the method)
  – Implementation details
    • Pseudocode not sufficient (Thimbleby, 2003)

• Software is an add-on
• Software should be required – it’s part of the scientific process
What’s the ideal?

- Can we define a gold (or minimum) standard?
- Case study – HCl
Software’s role in research

• Software is the tool that enables the research

• Software is enabling research in a new way or to a new group
  – So it is used for doing research AND is part of the research

• Software is the research
Software is...

...a tool to enable research

...the research

Upfront

Developed

Continuously

Retrospectively

Becomes an output

Continuously
Software is... (Computer Scientist)

...a tool to enable research

...the research

Upfront

Developed

Continuously

Retrospectively

Becomes an output

Continuously

Workflow management tools
Software is... (Bioinformatician)

...a tool to enable research

...the research

Upfront

Developed

Continuously

Retrospectively

Becomes an output

Continuously

Workflow management tools
Software is... (Biologist)

...a tool to enable research

...the research

Workflow management tools

Upfront

Developed

Continuously

Retrospectively

Becomes an output

Continuously
Software is... (Computer Scientist)

...a tool to enable research

...the research

Reasoners

Upfront

Developed

Continuously

Retrospectively

Becomes an output

Continuously
Software is... (Ontology author)

...a tool to enable research

...the research

Reasoners

Upfront

Developed

Continuously

Retrospectively

Becomes an output

Continuously
Software is... (Computer Scientist)

...a tool to enable research

...the research

Data collection app

Upfront

Developed

Continuously

Retrospectively

Becomes an output

Continuously
Software is... (Humanist)

...a tool to enable research

...the research

Upfront

Retrospectively

Developed

Becomes an output

Continuously

Continuously

Data collection app
Software is... (Computer Scientist)

...a tool to enable research

...the research

Analysis scripts

Upfront Developed Continuously

Retrospectively Becomes an output Continuously
Software is... (Clinician)

...a tool to enable research

...the research

Analysis scripts

Upfront

Developed

Continuously

Retrospectively

Becomes an output

Continuously
CHI 2016 best papers

• 23 papers
  – Pseudocode
  – Full analysis results of analysis
  – Data, source code
  – Tool, source code
Example 1

• Can we predict if someone will view an item of dynamically updating web content?
Analysis process

- Step through the video
  - Manually define areas of interest
  - Log start and end times

Final analysis: CHAID in SPSS – all data online.
Predicting whether users view dynamic content on the World Wide Web.

CAROLINE JAY, University of Manchester, UK
ANDY BROWN, University of Manchester, UK
SIMON HARPER, University of Manchester, UK

Dynamic micro-content—interactive or updating widgets and features—is now widely used on the Web, but there is little understanding of how people allocate attention to it. In this paper we present the results of an eye tracking investigation examining how the nature of dynamic micro-content influences whether or not the user views it. We propose and validate the Dynamic Update Viewing-likelihood (DUV) model, a Chi-Squared Automatic Interaction Detector (CHAID) model that predicts with around 80% accuracy whether users view dynamic updates as a function of how they are initiated, their size and their duration. The model is constructed with data from live websites and does not rely on knowledge of the user’s task to make its predictions, giving it a high-level of external validity. We discuss one example of its application: informing how dynamic content should be presented in audio via assistive technology for people with visual impairments.

7 The raw data used for the analysis are available at http://wel-eprints.cs.manchester.ac.uk/49/.
8http://www.slideshare.net/AcuityETS/tobii-studio-quick-start-guide

Safari Can’t Open the Page

Safari can’t open the page “wel-eprints.cs.manchester.ac.uk/49/” because the server where this page is located isn’t responding.
Example 2

- 100,000 fossils sitting in boxes
- Photographed by volunteers
- Can we catalogue them through crowdsourcing?
HCI research question

• Should people be required to register before they can contribute to a study?
• No – but they should have the option.

Extending the platform
Example 3

- [link](http://idinteraction.cs.manchester.ac.uk)
- SSI blog post: [link](http://goo.gl/kNVtFt)

Funded by EPSRC grant EP/M017133/1
IDInteraction
Software as academic output

- Software citation
- Credit
- License

- Software isn’t a separate thing
  - Software could exist without the paper
  - Paper couldn’t exist without the software
Software as Academic Output

• Computational research should be:
  – Findable
  – Accessible
  – Interoperable
  – Reusable

• Academic software should be:
  – Findable
  – Accessible
  – Reusable
  – Citable?
  – Extensible

And probably other things... This is a starting point.
Video processing: The Horror

• Install Linux (Ubuntu 14.04.2)
• Install mencoder (version 1.1-4.8)
  – Plus all 33 of its dependencies
    • And all of their dependencies, and so on...
• Run, for every input file, one after another:

  $ mencoder <input> -vf crop=640:360:640:0 -nosound -ovc x264 -o <front-view-output>
  $ mencoder <input> -vf crop=640:360:0:0 -nosound -ovc x264 -o <side-view-output>
  $ mencoder <input> -vf crop=640:360:640:360 -nosound -ovc x264 -o <rear-view-output>
Video processing: IDInteraction

• Install docker
  – Works on Windows, OS X or Linux

• Run, once:

  $ docker run -it -v <input-dir> -v <output-dir> idinteraction/video

  – This downloads the video processing pipeline and runs it over all of the input data automatically

• Come back tomorrow

  – For new data, just re-run the docker command above and it will only process new inputs
Object tracking: The Horror

• Assuming that Linux is already installed
• Install OpenCV (version 2.4.8)
  – Plus all 34 of its dependencies
    • Plus all of the development libraries and header files
    • And all of their dependencies, and so on...
• Install C++ compiler
• Install CMake
• Install Git
• Install CppMT (C++ version of CMT tracker)
  – Clone from GitHub
    • Get the IDInteraction specific version
  – Configure the build
  – Compile to produce ‘cmt’ executable
• Run, for every output of the video processing stage, one after another:
  $ cmt --bbox=<initial bounding box> --output <output dir> <video file>
Object tracking: IDInteraction

• Assuming that docker is already installed
• Run, once:

```
$ docker run -it -v <input-dir> -v <output-dir> idinteraction/tracking
```

– This downloads the object tracking pipeline and runs it over all of the input data automatically

• Come back tomorrow

– For new data, just re-run the docker command above and it will only process new inputs
Open source software

• Specify coding start point
• Outputs data into files
• Batch process videos
• Define, adjust and edit the bounding box

• Changes merged back
Making software available

- Package (exe, Ruby gem, Python egg)
- Docker image, virtual machine
- Online service
- Source code
To make it easier for ourselves and other investigators to use our automated pipelines, we have used software containers to wrap up each stage of our method with everything needed for it to run. This guarantees that our tools will always run the same way, regardless of the environment they are running in, and ensures that any research is as reproducible as possible.
The trouble with pseudo code

Thimbleby (2003) has illustrated an example which explains the problem of pseudo code and why it is not suitable to be presented instead of source code in scientific papers. In the case of Porter’s stemming algorithm which was invented to find the canonical form of any word, for example programmable and programming words will both refer to “program” word. The problem of Porter’s algorithm is that, it was published in a nonprogrammable form thus many incorrect implementations were produced based on that form. Porter himself recognized misunderstanding as one of his algorithm’s problems. Consequently, many researchers have presented incorrect results by applying an incorrect version of Porter’s stemming algorithm in their published work. Regarding this, Thimbleby (2003) believes that it is much easier to provide the source code on a website, for example, rather than trying to extract a description or pseudo code from the source code.