Video In PubPub: Moving Images in Context

by

Thariq Shihipar

BASc Engineering Science, University of Toronto (2013)

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Media Arts and Sciences

at the

Massachusetts Institute of Technology

September 2016

Copyright Massachusetts Institute of Technology 2016. All rights reserved.

Signature redacted

Thariq Shihipar
Program in Media Arts and Sciences
August 20th, 2016

Certified by

Andrew B Lippman
Senior Research Scientist, Program in Media Arts and Sciences
Thesis Supervisor

Accepted by

Pattie Maes
Academic Head, Program in Media Arts and Sciences
Video In PubPub: Moving Images in Context
by
Thariq Shihipar

The following served as a reader for this thesis:

[Signature redacted]
Thesis Reader

Danny Hillis
Visiting Professor, MIT Media Lab
The following served as a reader for this thesis:

Signature redacted

Thesis Reader .................................................................

Jason Foster
Associate Professor of Engineering Design, University of Toronto
Video has become so easy to make and distribute that it is becoming the most common way to describe many processes and ideas. However, it is often not effective at doing so. Within a context of a larger document, video is rarely a good citizen compared to text or images. It takes over the users attention and rarely points to other parts in the document. The video itself may be poorly made or it may demand too much time and investment on the part of the viewer relative to the content it provides. This thesis attempts to tackle this problem for a specific but useful circumstance: academic publishing.

In this thesis, I develop and study two novel methods of using interactive video within an academic document for instruction and peer review.

StepbyStep explores how we can create a better tool for instruction and documentation of methods. A key property of good documentation is that it is easily skimmable for relevant information. StepByStep breaks down a video into steps and arranges them spatially so that users can skim and scroll through the steps to find what they’re looking for.

VideoComments is an attempt to create an alternative method of leaving feedback to a document. By pairing a video recording of a user along with their actions on the document, we allow users to reference images and embeds easily, quickly switch between different parts of the document and record their thoughts in real-time.

Finally, this thesis creates an infrastructure in which interactive video applications such as StepByStep and VideoComments can be integrated into academic publishing standards and platforms.

**Thesis Supervisor:** Andrew Lippman  
**Title:** Senior Research Scientist, Program in Media Arts and Sciences  
Program in Media Arts and Sciences
Acknowledgements

Bismillah. In the name of God, the most gracious, the most merciful.

I would like to thank my family for their love and support. To my siblings, I can’t wait to see what you do; footsteps are meant to be swept over, not followed. To my parents, thank you for your sacrifices, for living in a place where you will always be foreigners and yet making a home.

I would not be here if not for my advisor, Andy Lippman. When I first met him, Andy told me that at the Media Lab we do our best work in fields we know nothing about. I spent the next two years realizing just how many things I knew nothing about.

In particular, I knew nothing about academic publishing. I would like to thank everyone who has contributed to make PubPub what it is so far, I hope this acknowledgement will mean much more in a few years. Travis, for being a tireless force of nature and doing it with a smile on his face. Hassan, for crossing the country to work with us, and having contributed so much so soon. The many people at MIT that support, trust and enable us to work on PubPub: Chris Burg, Ellen Finnie, Amy Brand, Nick Lindsay, Natalie Saltiel, Kevin Esvelt, Gabriella Gomez-Mont, Daniel Tello, Kevin Slavin, Neri Oxman and Joi Ito. And finally, the UROPS and others who contributed to PubPub: Mark Chounlakone, Emmanuel Oquendo, Abigail Russel, Jenny Xue, Ethan Bian, Dang Pham, Joel Gustav and Drew Shapiro. That last sentence could also be titled, ‘all the people who are smarter than me’.

I would also like to thank the rest of my research group, Viral Communications. Tomer, Ariel, Savannah, Vivian, Brian, Amir and Jasmine. They say that the best way to solve your own problems is to help someone else, that must be how we get so much done.

Thank you to my friends, Guru, Hisham, Kevin, Fareeha, Haniya, Bianca, Nabeel and everyone else. There are so many who live in this city but never make it their home, I am thankful to not being amongst them.

A special thank you to Linda, Keira, Monica, Amanda and Deb. The Media Lab is
just different from other places, I always tell people. There is such a great community here and you are all a vital part of that.

Thank you to all those who helped me get here. Ron Baecker, for taking a chance on me so many years ago and for writing me that recommendation letter not so many years ago. Nick Shim, for teaching me about both coding and life. Jason Foster, for showing me what it means to be an engineer. The many anonymous people on the internet forums that I grew up on and learned how to code from, you will likely never read this but I am forever grateful for those experiences.

And finally, thank you to Tayyab and the football crew for breaking my thumb with that touchdown pass. The only thing I couldn’t do in my cast was play video games and that’s probably the reason this was finished. Sorry you couldn’t win a game without me.
# Contents

This document was written on PubPub and can be read online at:
https://www.pubpub.org/pub/video-in-pubpub

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Introduction</td>
<td>9</td>
</tr>
<tr>
<td>1.1 - Purpose</td>
<td>10</td>
</tr>
<tr>
<td>1.2 - Background &amp; Motivation</td>
<td>11</td>
</tr>
<tr>
<td>2 - Background &amp; Motivation</td>
<td>12</td>
</tr>
<tr>
<td>2.1 - The PubPub Platform</td>
<td>13</td>
</tr>
<tr>
<td>2.2 - Interactive Video</td>
<td>14</td>
</tr>
<tr>
<td>2.2.1 - Distribution &amp; Editing</td>
<td>15</td>
</tr>
<tr>
<td>2.3 - Interactive Video for Learning</td>
<td>19</td>
</tr>
<tr>
<td>2.4 - Instructional Video in Science &amp; Reproducibility</td>
<td>21</td>
</tr>
<tr>
<td>2.5 - Video as a Method of Feedback</td>
<td>23</td>
</tr>
<tr>
<td>3 - Architecture of Interactivity &amp; Video</td>
<td>24</td>
</tr>
<tr>
<td>3.0 - Goals</td>
<td>25</td>
</tr>
<tr>
<td>3.1 - PubPub Methodology</td>
<td>25</td>
</tr>
<tr>
<td>3.1.1 - Rendering Pipelines</td>
<td>25</td>
</tr>
<tr>
<td>3.1.2 - Issuing DOIs</td>
<td>25</td>
</tr>
<tr>
<td>3.2 - Design Guidelines For Interactivity &amp; Video In PubPub</td>
<td>28</td>
</tr>
<tr>
<td>3.3 - Architecture: Assets &amp; Atoms</td>
<td>30</td>
</tr>
<tr>
<td>3.3.1 - Database Schema</td>
<td>31</td>
</tr>
<tr>
<td>3.3.2 - AtomType Architecture</td>
<td>32</td>
</tr>
<tr>
<td>3.4 - Approach: Video Implementation</td>
<td>34</td>
</tr>
<tr>
<td>3.5 - Results &amp; Future Work</td>
<td>34</td>
</tr>
<tr>
<td>4 - StepByStep</td>
<td>36</td>
</tr>
<tr>
<td>4.0 - Goals</td>
<td>37</td>
</tr>
<tr>
<td>4.1 - Related Work</td>
<td>37</td>
</tr>
<tr>
<td>4.2 - Design Guidelines</td>
<td>38</td>
</tr>
<tr>
<td>4.3 - Architecture</td>
<td>40</td>
</tr>
<tr>
<td>4.4 - Approach</td>
<td>42</td>
</tr>
<tr>
<td>4.4.1 - User Interface</td>
<td>42</td>
</tr>
<tr>
<td>4.4.2 - Viewing</td>
<td>42</td>
</tr>
<tr>
<td>4.4.3 - Authoring</td>
<td>43</td>
</tr>
<tr>
<td>4.5 - Evaluation</td>
<td>46</td>
</tr>
<tr>
<td>4.5.1 - Evaluation: Speed of Learning</td>
<td>46</td>
</tr>
<tr>
<td>4.5.2 - Metrics</td>
<td>46</td>
</tr>
<tr>
<td>4.5.3 - Questions</td>
<td>47</td>
</tr>
<tr>
<td>4.5.4 - Analysis</td>
<td>47</td>
</tr>
<tr>
<td>4.6 - Results &amp; Future Work</td>
<td>49</td>
</tr>
<tr>
<td>5 - Video Comments</td>
<td>51</td>
</tr>
<tr>
<td>5.0 - Goals</td>
<td>51</td>
</tr>
<tr>
<td>5.1 - Related Work</td>
<td>51</td>
</tr>
<tr>
<td>5.2 - Design Principles</td>
<td>52</td>
</tr>
<tr>
<td>5.3 - Approach</td>
<td>54</td>
</tr>
</tbody>
</table>
5.3.1 - Architecture .............................................................. 54
5.3.1.1 - Scaling & Reflow ................................................... 55
5.3.2 - Interface ................................................................. 56
5.3.3 - Authoring ............................................................... 56
5.3.4 - Viewing ................................................................. 57

5.4 - Results & Future Work .................................................... 58
5.4.1 - Possible User Study ................................................... 59

6 - Conclusion ........................................................................ 60

References ............................................................................. 61
1. Introduction

PubPub is a program within the Viral Communications group that addresses the transformation of academic publishing from a single-authored research checkpoint to a modern, collaborative process [1]. It includes components to allow creation of a document that has the following features:

- illustrations that are processes instead of snapshots
- inclusion of raw experimental data to permit reader-processing
- a set of named and volunteer reviewers
- an editing process that favors development

PubPub seeks to solve some of the problems in publishing that have plagued academic authors. In particular, the characteristic of much modern science is that it is collaborative and evolving. Often collaborations span traditional academic domains and collaborators may contribute words, data, code, images, videos, methodologies or almost anything else. However, due to the difficulty of representing this work in static documents, these contributions may not be included in the paper or stuffed into appendices.

PubPub addresses this by enhancing the capabilities of standard academic academic documents to include interactivity and dynamism. In PubPub, the work is part of the document, code, graphs and data live alongside text as embeds instead of being portals as links. The goal and guiding principle of interactivity in PubPub is to create more efficient knowledge sharing interfaces, not simply novelties or tech demonstrations[2].

Video is a particularly interesting part of the dynamic capabilities of PubPub, because so much of the way we document our world is through video. In PubPub, the simplest form of video may involve embedding a youtube video in a document, as opposed to including it in the accompanying files.
However, the nature of linear video is that often apart from the context it lives in: a remnant of static documents. You can either watch the video or read the document, it behaves more like a link than an embedded resource. Separation of video from the document has been found to be detrimental to content learning and engagement[3][4].

In this thesis I explore integrating video tied to the document with the goal of more effective knowledge sharing. Moreover I aim to explore one area in which video is poorly but commonly used, and another area where video is rarely used but could be used to great effect. These use cases are instructional learning and commenting/peer review respectively.

Instructional learning is often intensely visual and best aided by moving images, but linear video separates it from the document in which it lives. By dividing the video into steps and integrating it into the document, I believe that users will be more likely to engage with the content and learn the content more deeply and quickly.

Peer review on the other hand is almost entirely text-based, digital verbal peer review suffers from an inability to interact with the document. By linking the video and document, we aim to create effective verbal peer review that points to and interacts with the document. I believe that these Video Comments have the potential to make commenting on interactive media more effective and/or reduce the time burden of reviewing for reviewers.

1.1 Purpose

This thesis proposes an architecture for interactive video within a modern publishing platform and investigation into interactive video as a solution to these problems within traditional publishing.

The focus of the background research is on scholarly publication, but our hope is that the systems of scholarly publishing, when well implemented and modernized, can be used to share knowledge in much broader circles that simply academia. Uses for PubPub can range from DIY, cooking, fabrication,
sports drills, video game tutorials, government legislation and more. We believe that PubPub can accommodate the needs and interests of these groups without sacrificing its focus on scholars[1].

1.2 Contribution & Goals

This thesis has three primary contributions.

First, I explore an architecture for interactivity within PubPub that enables interactive video while still maintaining the requirements for an open access scholarly journal. I aim to create an architecture that fulfills the requirements of the Open Access Scholarly Publications Association so that it acts as a stable base upon which to build interactive video in PubPub.

Second, I create and evaluate StepByStep, an interactive video system which breaks down a video of a process into steps and arranges them spatially so that users can skim, scroll to and repeat the relevant parts of the video, adjusting as they learn. I aim to show that StepByStep increases knowledge transfer by increasing engagement and learning.

Third, I create and evaluate Video Comments, an interactive video system which is an attempt to create an alternative method of leaving feedback to a document. By pairing a video recording of a user along with their actions on the document, we allow users to reference images and embeds easily, quickly switch between different parts of the document and record their thoughts in real-time. I aim to show, first that video is a viable commenting interface, that it is technically feasible and intuitive to navigate. Second, I aim to show that Video Comments offers a meaningful alternative to text, that it either produces similar quality comments or it creates new types of comments that are not possible using text.
2. Background & Motivation

In this section, I aim to tie together background research in video, interactivity and scholarly publications in order to form a coherent base upon which to introduce the work featured in this thesis.

I begin by giving a description of the PubPub platform and its capabilities, with an eye towards interactive and non-linear video.

I then give a broad overview of research into interactive and non-linear video with a focus on their applications for learning and their advantages over linear video.

Next, I introduce two areas of scholarly publications in which interactive video may be useful: documentation of methods and peer reviews of publications.

Regarding documentation of methods, I outline the increasing use video to document processes within a scholarly publications. I posit that because the point of videos within publications is to teach, we can utilize interactive video to increase effectiveness and engagement.

With respect to peer review, I show a trend towards open methods of peer review in scholarly publications. By removing the hard constraint of blind review, scholarly peer review more closely resembles other methods of peer assessment. In peer assessment literature, verbal communication is often used and leads to improving review when used in conjunction with text. I suggest that interactive video could be used as a method of providing verbal feedback in online scholarly peer review.
2.1 The PubPub Platform


PubPub is an end-to-end publishing system, so authors write their work in the same platform where they publish it. Further, commenters and reviewers have access to the same writing tools that authors do.

One of the most powerful parts of the PubPub writing tools are ‘embeds’. Traditional publishing platforms are limited to static images as embeds, but in PubPub the web is the primary mode of output. Almost anything on the web can be embedded and made interactive, from graphs to videos.

Thus, PubPub has the ability to introduce new, non-standard interfaces (such as live data visualization or interactive video) and not immediately worry about interopability issues, each PubPub client carries the code to render and edit all interactive elements within a document.

This deep entanglement between document and embeds in PubPub I believe offers a solution to the key problems found in interactive video: distribution & editing.
2.2 Interactive Video

The appeal of interactive video is clear, why should we not be able to interact with moving pictures in the way we interact with text? In fact, a large body of work has been put into researching interactive video & its editing interfaces[6][7][8].

However the primary challenges in building such interfaces, as noted by Meixner et. al [7] is the lack of structure in interactive & non-linear video as compared with linear video. Linear video has a standardized implementation: a scroll-bar, play and pause buttons, with possible options such as fast-forward and reverse. There there are no agreed standards for implementation of non-linear video interfaces, despite a number of standards proposed such as MHEG[9].

In fact, as Mexiner et. al [7] point out in the diagram below, non-linear video can sometimes be confused with interactive video, annotated video and hyper video, all of which have slightly different features, use cases and possible implementations.
Mexiner et. al [7] found a large divergence in the definitions of interactive video, however they found the most common shared elements were: alternative playback paths, an influence on the order of scenes and choice elements were elements that created an interactive video.

This creates the twin problems of **platform distribution** and **editing interfaces**. Tools that create interactive nonlinear video must make a tradeoff between the two.

### 2.2.1 Distribution & Editing

Interactive video solutions often take one of two paths: fully-featured or specific use-case.

Fully featured editors optimize for platform distribution. Having a standard format allows distribution of well-defined code to devices that can then be used to play almost any sort of interactive video within the standard, one such
proposed standard was MHEG[9]. This also creates a standard communication protocol that the rest of the document can use to communicate with the video. The complexity is shifted on to the author, who must carefully produce both the video and the interactions which govern it.

Specific use-case editors optimize for simple editing interfaces. The author is given a formula for the type of video they want to make, for example a recipe and must simply produce the video and make it fit the format.

Some examples of a generalized fully-featured editor are AthenaMuse[6] or SIVA player[7]. These tools can be broken down into a few components:

- **Scene Creation** - Dividing a video into discrete scenes either manually or aided by an algorithm.

- **Frame Editing** - Manipulating pixels in an individual frame or set of frames, to add effects, remove distortion, etc.

- **Scene Graph** - Connecting different scenes together in some graph-like fashion to create an order of viewing.

- **Annotation Layer** - Annotating frames or scenes with text, simple drawings, etc.

However, given the complexity of these components, general interactive video editors are often unwieldy (pictured below) and there are few examples in popular consumer products. In fact, most alternate video editors have instead simplified by forcing hard constraints on the user, such as Vine or Instagram’s boomerang. These video interfaces exist as part of a larger document such as a newsfeed.
An example of SIVA player – a generalized nonlinear video editor.

In contrast, Instagram's boomerang video editing feature focuses on extreme simplicity and a single use-case.

PubPub being an end-to-end platform, however simplifies the distribution problem. PubPub can support any number of small, niche interactive video interfaces and instantly share them with all PubPub clients. Thus, this shifts the focus to creating simple, but effective editing interfaces.

Given our focus on scholarly publishing, we are particularly interested in video
as it relates to knowledge and instruction. How can we optimize interactive video for meaningful learning?
2.3 Interactive Video for Learning

Research into interactive video often features educational video as its killer application [10] [3]. Mayer and Marino[4] offer a mental model of why this may be the case. They suggest that the human information-processing system consists of two separate channels— visual and auditory/verbal. In multimedia these channels are fed by inputs that are either words, pictures or some combination of the two.

Mayer and Marino [11] note meaningful learning requires that learners engage in substantive cognitive processing in the verbal and visual channels. Informational overload occurs when the cognitive processing required exceeds the learner’s capacity. This cognitive processing divide this processing into two parts: essential and incidental. Essential processing is inherent to the material being learned whereas incidental processing is not required by the material but rather primed by the design of the task.

![Diagram of multimedia presentation and memory processes]

Mayer and Marino's cognitive theory of multimedia learning.

In order to reduce the amount of incidental processing required by video for meaningful learning, Mayer and Marino [?] suggest a variety of load-reducing methods, such as:

- **Segmenting** - Allow time between bit-sized segments.

- **Weeding** - Eliminate interesting but extraneous material to reduce processing of extraneous material.
• **Signaling** - Provide cues for how to process the material to reduce processing of extraneous material.

• **Aligning** - Place printed words near corresponding parts of graphics to reduce need for visual scanning.

It is clear that linear video alone cannot implement such load-reducing methods, interactivity and non-linearity is needed.

This is shown experimentally as well, Zhang et al.[12] found that interactive, non-linear video when integrated into an e-learning document improves learning effectiveness as compared with linear video or no video. McKay and Davenport [8] cite 'synchronization of media' (weeding, aligning) and 'user control of dimensions' (segmenting, signaling) as the key requirements in creating interactive educational software.

Yoder[3] found that the effectiveness of interactive video in e-learning depends on the type of learner they are (passive vs. active), with passive learners preferring linear video and active learners preferring interactive video. Matching this, Mayer and Marino[4] found that different learners have different cognitive processing and channel capacities.

Overall, it is clear that interactive video embedded into a document is more effective for meaningful learning than linear video or only text. In some areas of publishing this is desperately needed, because text and images are simply not enough, one such critical area is documentation of methods in science.
2.4 Instructional Video in Science & Reproducibility

As described by Pasquali [13] scientists have been using video to document their work for over 100 years. One of the first scientific films was by Etienne-Jules Marey who built a ‘photographic rifle’, a proto-camera that was built like a rifle but captured 12 frames per second, to film the flight of seagulls. As video has become increasingly common in mainstream media, its use by scientists has increased. Scientific journals increasingly provide supplemental video files along with papers in order to show information that can’t be communicated by text and still images alone, often videos of the results or of the experiment being run. Often these videos maybe promotional and be used to generate excitement for the paper at a conference, as is often the case at CHI[14].

However, Pasquali [15] notes that experimental protocols and methods are an excellent use for video, in order to communicate the exact procedure along with text. Pasquali also addresses the obvious objection:

Readers may object that they don’t have time to set up Hollywood studios to publish a paper. But the effort required for filming a relevant procedure is outweighed by the fact that more precise description of methods greatly aids scientific understanding and facilitates future research. Furthermore, videos are valuable tools for teaching.

This has not gone unnoticed in the scientific community and as early as 2006, websites such as Nature Protocols, Cold Spring Harbor Protocols, OpenWetWare and DNAtube offered both the software and the server space for hosting movies of protocols [16]. Perhaps the most promising of these initiatives is Journal of Visual Experiments, [17] founded in 2007. For every accepted submission to JoVE, a video crew is sent to record step-by-step video instructions for reproducibility and pointing out possible technical problems.
Besides clarity of instruction, Pasquali notes two other benefits to video instructions. First, allowing experimenters to see their experiment from a third party, known as the external observer condition which Aloe & Kristen[18] cite as being crucial for science.

The second being improving peer review, Pasquali notes that video documentation of methods then opens up more in-depth review of the procedure. If done widely, this help could contain fraud. For example, the famous stem-cell cloning fraud involving Korean scientist Woo-Suk Hwang was first uncovered by an anonymous Korean biologist inspecting some of the photographs in Hwang’s Science paper[19].

However, actual use of instructional video to document process is limited by the format in which papers are presented, namely PDFs or at best static HTML. As noted in Section 2.2, interactive video is limited by the platform that is used and scientific paper platforms currently lack any sort of support beyond even the simplest linear videos. In Section 4, I outline how PubPub can change this through an interface called StepByStep.
2.5 Video as a Method of Feedback

During scholarly peer review, a group of external experts evaluates a work to determine if it should be published. In order to reduce effects of personal bias, the identity of the reviews and authors are kept secret in what is called double-blind peer review. However, the effectiveness of blind peer review has recently been called into question[20][21].

Travis Rich[2] notes that instead of a single, pre-publication blind peer review academia may benefit more from a continuous, open peer review. This is one of the key experiments of PubPub[1].

By not requiring blindness, peer review more closely resembles other methods of peer assessment in workspace and classrooms. Optimal peer assessment in these contexts often consists of both oral and written feedback[22][23][24].

Van de Berh et al. show that oral and written peer assessment focus on different parts of feedback. Oral feedback contains more explanation and feedback, and focuses more on style than structure [24][25].

However, oral feedback is currently missing from the formal scientific peer review process. In Section 5, I outline a prototype of audio-visual feedback in PubPub called Video Comments.
3. Architecture of Interactivity & Video

This chapter describes the challenges of utilizing video (interactive/non-linear or not) in a scholarly platform and describes the architecture that PubPub uses to realize this and other interactive elements.

As a modern, online publishing platform, PubPub aims to expand the capabilities of traditional scholarly documents. However, this increase in complexity must be carefully managed so as to maintain one of the ideals of scholarly publishing: preservation of knowledge for the future.

Code and bits are susceptible to rot as time goes on and new devices, new interfaces and new technologies come into play. How can we design the interactivity of PubPub to take advantage of all the affordances of modern devices but allow it to be easily adapted to the possibilities of the future? In particular, what does this architecture look like for linear and interactive video within PubPub?
3.0 Goals

The goal of this chapter is to outline a standards compliant architecture upon which interactive elements, specifically video, can be built. There is no single, unified definition but the Open Access Scholarly Publications Association (OASPA) [26] provides a thorough one which we will use as a standard.

The OASPA lists nearly 20 requirements, most of which are not relevant to interactivity. The two key requirements are: a plan for backup and the availability of Digital Object Identifiers (DOIs). A plan for backup requires that all content in the system should be easily backed up and hosted by external services. DOIs (explained in Section 3.1.2) refer to permanent ids to which a publication can be retrieved from and attributed to.

These two goals drive the architecture of interactivity and video within PubPub.

3.1 PubPub Methodology

In order to introduce the proposed architecture, some concepts within the PubPub codebase must first be introduced.

3.1.1 Rendering Pipelines

PubPub uses a model-view architecture for rendering. Data per pub is stored in a generalized JSON format which can then be processed by renderers to create output. The primary output is HTML, but other possible output examples include RSS, PDF or ePub. This means that every extension to PubPub must include both a model of how the data works and various renderers for different outputs, but it allows new renderers to be built so that new technologies do not render pubs obsolete.

3.1.2 Issuing DOIs

In PubPub each ‘embeddable’ work, essentially anything that is not text, can be issued a Digital Object Identifiers (DOI). This allows for individual contributions,
to graph, code, videos, etc. to be cited and granted credit for.

The DOI system as defined by [27] is a managed system for persistent identification of content on digital network. The DOI system is implemented through a federation of registration agencies, under policies and common infrastructure provided by the International DOI Foundation (IDF) which developed and controls the system.

A DOI is a string which is resolve to a resource. A DOI can resolve to any identifier, such as a ISBN or URL. Some example DOIs are:

10.1234/NP5678
10.2224/2004-10-ISO-DOI

Thus, in a scholarly setting a DOI can point to an online URL, a paper book, a newspaper, etc. Further, it gives a unique identifier to attribute citation to, instead of depending on URLs or bibliography matching [28].
As such, DOIs have become widely adopted in Academia, forming a key feature of the CrossRef system.

In a way DOIs have become the unit of scholarly publication, they are cited and shared, but critically a DOI is not limited to a paper. DOIs may be issued to datasets, graphs, code, images, or any other contribution [29]. Some startups such as FigShare [30] have been created to do just that, allow scientists to cite their work.
3.2 Design Guidelines For Interactivity & Video In PubPub

In PubPub, video (whether interactive or linear) is used in a document via an ‘embed’. Each embed can be issued a DOI such that individual works can be cited and contributed. This is a fundamental part of realizing PubPub’s author driven philosophy[1].

However, this in-turn creates constraints on all interactivity within PubPub to be required to fit into the DOI Format. This format can be boiled into four key points: uniqueness, resolution, interoperability and persistence.

Uniqueness (1 resource has only 1 DOI) and resolution (the process of turning a DOI into a resources such as url, barcode, etc) are handled mostly by the DOI system.

However, interoperability and persistence are relevant to this thesis. Interopability means that it must work across computers and networks, moreover PubPub takes this to mean that it must be able easy to embed in other works. For example, a proprietary image format may be faster and save space but a standard one will allow others to easily embed it in their work. For any non-standard implementations, the viewer must bundled come with the document, and be open sourced.

In Section 2.2, we discussed how linear video is given an inherit advantage by its standard playing interface. This once again arises as a challenge for interactive video from a publishing perspective.

Persistence means that each DOI represents the same reference forever, this may be thought of as ‘interoperability with the future’[27]. From the perspective of interactive video this means that any technology developed must be able to easily be ported to future platforms and devices.

Overall, when using video and other interactivity in PubPub we optimize for
transparency and interoperability over efficiency.

Through this we come to realize a key design guideline: Interactive components (video in particular) should be broken down into a few key components: source data, interactive meta-data, viewer code. This unbundling means that the viewer code may be easily updated to maintain persistence or switched out for interoperability whereas the unchanging metadata and source data will define the behavior.

For interactive video, this means that all source files must stored as video in PubPub, even if all the video is not utilized and could be cut down for efficiency. The meta-data stores the transformations on the source data that must be done to make them interactive, e.g. the loops and sections that they focus on. The code to display and manipulate the interactive video must be entirely open sourced (GNU or MIT license) and at most depend on standard HTML APIs.
3.3 Architecture: Assets & Atoms

In PubPub we realize these design guidelines with a model representation using Atoms, AtomTypes, Assets & Versions.

Assets are anything that the user may upload to the PubPub servers and can be represented as a single URL. This may include images, videos, datasets, or even files that PubPub does not recognize. Assets are to be treated as permanent links, a link that will always return the same file and never be removed as long as PubPub is functional.

Atoms are anything that can be given a DOI in PubPub, essentially the atomic unit of publishing. Atoms can be pubs, discussions, images, videos, datasets, graphs, code or anything else that can be published. Atoms contain assets (though this is not a one-to-one relationship) and are versioned.

Atoms each have an AtomType which defines the type of data stored in its versions and how it is displayed. AtomTypes each come with corresponding code for editing interfaces (pictured below) and viewing interfaces. Adding new AtomTypes requires adding to the main PubPub codebase.

Versions are immutable, individual publications of each Atom. DOIs are assigned to individual versions, not Atoms. Versions contain a data field which is completely defined by the type of Atom that they represent. Typically it will include at least one asset url and some meta-data used for interactivity and display. Updates to an atom result in a new version being created. Some sample contents of a version object are included below.

All of these models are all stored in Javascript Object Notation (JSON) format, which has become the defacto format for storage and communication across the web.
3.3.1 Database Schema

To implement these models, our schema represents relationships (embeds, parent-child) between Atoms as Links and updates to Atoms are stored directly on the model as Versions.

For example, a Pub may have several versions, which may all embed different other Atoms (EmbedAtoms). This relationship between a Pub Atom and an Embed Atom is stored in a Link object.

This database schema is pictured below:
3.3.2 AtomType Architecture

Each AtomType consists of a few key components:

- Editor - Takes in assets or existing atoms, allows the user to edit them and saves them to the server.

- Viewer - Takes in a version of an atom and renders it to HTML.

- Version Content - The structure of the content field in the version object.

- Asset Types - The filetypes of assets associated with this AtomType.

The relationship between them is illustrated below:

The base AtomTypes in PubPub are: Image, Video, LaTeX, PDF, iFrame, Jupyter Notebook and reference.
Each new AtomType must implement all four components. For example, a fully-featured Image Atom would consist of:

- **Image Viewer** - Outputs the image appropriately scaled, styled and captioned.
- **Image Editor** - Allows users to modify or replace the image.
- **Version Content** - Stores the asset url of the image along with metadata such as geolocation, source url, etc.
- **Asset Types** - All image files such as .png, .gif, jpg, .svg.
3.4 Approach: Video Implementation

By design, linear and interactive video fit neatly into this framework. Users upload video assets to PubPub either through the media library interface or, in the case of video comments, through their webcam. Video assets are stored in Amazon S3 storage, but urls are proxied through static.pubpub.org such that even if the location of files were to be migrated, the url could stay the same. In the future, large source videos that do not need to be downloaded often maybe stored in high retrieval latency storage such as Amazon Glacier[32].

Each type of video representation is an AtomType. Linear Video simply takes a video asset and displays it with a player. The version data stores the asset url and data on how to display the video relative to the rest of the pub: size, alignment, captions, etc.

The StepByStep AtomType takes a video and gives the user an interface with which to create loopable steps. The source video asset urls as well as the transformations required to create the steps are stored in the version data. The StepByStep AtomType is discussed in more detail in Section 4.3.

The VideoComments AtomType stores an uploaded video asset urls as well as a time series of actions on the document to be played back. The VideoComments AtomType is discussed in more detail in Section 5.3.

Overall, data is stored in standard video file formats, whereas all interactivity is described by the metadata and displayed by the AtomType code. Every video is able to be assigned a DOI and thus cited and shared. Both the format of data storage and code are open source so they may be adapted as browsers, devices and humans change.

3.5 Results & Future Work

In conclusion, the video architecture of PubPub appropriately meets the
standards of the Open Access Scholarly Publications Association[26]. By creating a separation of data and content backed by open standards and code, we minimize bit rot and allow for the use of new devices and filetypes without breaking changes. This creates a stable base upon which to build interactive video applications on PubPub.

In the future, the archiving system may be explored in-depth technically. While it is clear how to archive this data, there are technical challenges of long-term storage and handling of very large files.
4. StepByStep

This chapter describes the approach, design guidelines and methodology for StepByStep, an interface that creates interactive video to help document processes and skills.

As described in detail in Section 2.4, video is a useful way of documenting experimental protocols in science and more broadly, any skill or process to be learned.

However, linear video is a poor format for this, because it has no spatial dimensions; users must scrub through the timeline to find relevant information. For this reason, most embedded video, for example in the news, goes unread [33]. This creates a separation between the document and the video, the user can either be watching the video or reading the document, but not both. Compared to images, linear videos lack glanceability and skimmability in a document[34].

Animated GIFs have successfully been used as a happy medium between video and images. But animated gifs lack sound, play endlessly and have no coordination with each other. While useful for emotional content[35], their use is limited for knowledge. As I outlined in Section 2.3, chunking videos into segments aids knowledge transfer and decreases dropouts for educational videos [36].

In this section I design, develop and evaluate StepByStep, an alternative to linear video that chunks videos into segments and makes these segments part of the document. In this way, authors can describe their methods and process visually as part of the document.
4.0 Goals

The goal of StepByStep is to create an interface for instructional learning that is more effective at knowledge sharing than linear video. In particular, I aim to evaluate improvements in both engagement and learning. Are users more likely to engage with StepByStep videos than linear video and if they do, do they learn content more effectively and quickly than with linear video?

4.1 Related Work

JoVE[17], the Journal of Visual Experiments is a journal that sends video crews to capture the methodologies of the papers (primarily biological and chemical) that it accepts. While professionally edited and curated, in JoVE videos are the publication itself, a paper may be attached but the two are distinct entities.

Pavel et al. [36] created Video Digests, which breaks down a lecture into a series of video chunks associated with text, so that students can skim through a lecture to find relevant material. These videos however are often 5-10 minutes long and are the primary way of giving content, they may be thought of as a spoken paper divided into chapters.

Tiffany Tseng [37] made Spin, a platform for creating video that documents build processes for makers by forcing a hard constraint on the documentation. Spin consists of a turntable and recorder, and all progress is documented by spinning the work in progress on the spin table to generate a GIF.

A number of online GIF generators tools exists such as those made by Imgur, Giphy and Youtube. Savannah Niles[35] proposed work that would turn video into loopable segments so they are easily digested. However, she notes that GIFs and similar mediums are best for emotional content as opposed to informative.
4.2 Design Guidelines

A few key design guidelines were identified before the development of StepByStep:

- **Versatility in Output**

  The StepByStep codebase should be designed to be used as an external app, within the PubPub ecosystem or easily embedded in other contexts and apps.

- **Coherence with PubPub**

  While StepByStep is a versatile tool, its backend design should be driven by needs for PubPub and scholarly publishing. The design guidelines for the data structure and storage of interactive video within PubPub is described in **Section 3**. This I believe can be done without sacrificing versatility in output.

- **Flexible Input**

  StepByStep should work with video uploaded anywhere on the internet.

- **Precise, Simple Editing**

  The editing interface should be simple, with few buttons or controls, but allow for precision in the output.

- **Favor Simplicity Over Completeness**

  The taxonomy of processes is large and not all will be described by a sequence of videos. However, as described in **Section 2.2**, interactive video often falls prey to convoluted interfaces when attempting to be complete. For this attempt, we will prefer to be simple and focus on a common use case.

- **Default to Native Interactions Over Custom**

  Because users are not nearly as familiar with interactive video as with linear,
defaulting to OS and device native behaviors is preferable. For example, on the
desktop prefer scrolling to hiding elements behind buttons, on mobile, prefer
gestures to swipe between scenes.

- **Customizable ‘Modes’**

Because of the many uses and contexts for StepByStep, it is unlikely that one,
generalized interface will be sufficient. For example, the interface to learn how
to shoot a basketball on your phone is likely different from an interface to
show you how to do a precise scientific experiment on a desktop, lab computer.
For that reason, the look and interaction with the steps should be defined by
the mode of the output.
4.3 Architecture

StepByStep is built entirely in Javascript using React.js for modular frontend code, Node.js for backend APIs and MongoDB for data storage and retrieval. Both the frontend and backend use NPM [38] for package management and dependencies. All of StepByStep is hosted in the cloud on Amazon AWS.

Through NPM’s modular dependency structure, StepByStep can be used by itself, as an AtomType in PubPub or embedded using an iframe in another document.

As an AtomType in PubPub, StepByStep is defined primarily by video assets and a set of steps defined on them. This schema is illustrated below:
4.4 Approach

4.4.1 User Interface

StepByStep is divided into two interfaces, the authoring and viewing interfaces. The core concept of StepByStep is simple: break down a video into steps and arranges them spatially so that users can skim and scroll through the steps to find what they're looking for.

4.4.2 Viewing

Examples of StepByStep interfaces can be found at the following links:

- https://www.pubpub.org/pub/stepbystepdemo

Below is an animated example of the viewing interface for Step By Step.
Fisherman's knot

There is at any one time one **active** step, which is triggered by a user interaction or timing. By default it is by hover, but this change be changed by the 'mode' of the step In particular a mode can control two things:

- Appearance of steps - Border, numbering, sizing, etc
- Interactions with steps - Whether the active step is triggered through hovering, timing, clicking, etc. and the behavior of the non-active steps

In this way, modes allow customization to the genre of knowledge and the type of device it is viewed in.

4.4.3 Authoring

The standalone StepByStep authoring process can be tried out here:

http://pubpub-stepbystep.herokuapp.com

Authoring within StepByStep eschews complexity for interfaces that resembles
consumer products more than professional video editing services. Crucially, StepByStep does not require nor diminish professional video editing. Carefully edited videos still retain their polish whereas amateur video is not out of place as well.

The StepByStep authoring process also encourages authors to use videos that are not their own (assuming copyright issues are resolve) and annotate them to create an informative commentary.

Below is a breakdown of the authoring interface annotated with animations:

1. **Input Video**

   Users can drag video files to the input box or paste in video hosting URL such as youtube, vimeo, etc. The selected video will be downloaded and added to the library.

   ![An animated example of video uploading. This is best viewed on the online pub.](image)

2. **Video Selection**

   Users click on a video in their library to begin making a step from it.

3. **Step Brushing**
Using the brushes on the progress bar at the bottom, the user can slowly select and control the start and end points of a step. Fine grain controls to the right of the progress bar allow the selection of individual frames.

**Step By Step**

![Image of video brushing interface]

An animated example of video brushing. This is best viewed on the online pub.

4. **Step Options**

To the right of the video, the options for each step are displayed. The user can modify the playback speed for the step, whether it is muted, etc.

5. **Step Creation**

After finalizing the parameters of a step, it is added to the step arrangement area.

6. **Step Ordering**

Steps can be dragged to be ordered in a particular flow.

7. **Publishing**

After the content and order of steps is determined, the user can publish a view
4.5 Evaluation

Because the StepByStep authoring tools have not yet been merged into the live public PubPub ecosystem yet, we will focus on evaluation through a small scale, focused study. In particular, we focus on the speed at which Step By Step enables learning of a simple, motor skill.

4.5.1 Evaluation - Speed of Learning

The primary metrics we evaluate in this study are usability and speed. Given a simple, motor skill and a StepByStep interface, is a user able to learn how to complete the task accurately without becoming confused or blocked?

To evaluate this, the chosen skill was tying a simple, but uncommon knots. The two knots chosen were the figure-8 and the fisherman’s knot. For each knot, a youtube video made by howcast.com was selected and a StepByStep interface was made from it. Howcast was selected because they are a creator of ‘how-to’ youtube videos and explicitly label steps within their video to minimize differences between the two formats.

Video 1, the figure-8 is a fairly simple knot that most people, with attention can complete without pausing the youtube video.

Video 2, the fisherman’s knot, often takes several pauses and backtracks while using youtube.

Participants were asked to execute one knot using only the youtube tutorial, and then asked to tie the other knot using only the StepByStep interface. Participants were randomized such that half would use the the youtube interface for a knot and the other half would use the stepbystep interface for the same knot.

4.5.2 Metrics

The following metrics were recorded per participant:
1. Time - The time it takes them to perform the tie the knot in seconds.

2. Scaled Time - The time it takes them to perform the task, relative to the linear video time in seconds.

3. Backs - the number of times the user moved back in the process to relearn a step.

4. Startovers - the number of times the user failed to do the task and had to start over from scratch

5. Heatmap (StepByStep only) - A JSON formatted object which stores the number of times each step was hovered and looped over.

**4.5.3 Questions**

After completing the evaluation, users were asked the following questions.
Between StepByStep and Youtube which one would you prefer for:

1. Speed of completing the task

2. Usability of the interface

3. Pleasantness of the interface

4. Clarity of the instructions

5. Teaching someone else

**4.5.2 Analysis**

6 Participants were recruited from amongst MIT grad students to complete the evaluation. Each one took on average roughly 15-20 minutes. The raw data and calculations are embedded below.

Raw data & derivations are best seen online, the full google spreadsheet can be seen at: https://goo.gl/3FeryR

From this data, the following findings were distilled:
1) **Faster** - The StepByStep interface on average caused the task to be completed 33% faster than the youtube interface. Qualitatively, 5 out of 6 users indicated that they thought the StepByStep interface was faster to learn from.

![StepByStep vs Youtube -- Time to Complete Task](image)

2) **More Forgiving** - In StepByStep backtracks were less correlated with increasing task time. For example, having to go revisit previous steps two times would double the time it took to complete the task using the Youtube interface, but only increase it by 50% using StepByStep.

![StepByStep - Scaled Time vs. Backs](image)
3) **Less Familiar** - A common comment amongst participants was that they found the Youtube interface to be more familiar, which led them to rank it more clear or more pleasant than StepByStep.

4) **Lacking Features** - Participants identified the need for meaningful pauses and the ability to see the duration of each step in order to anticipate key moments in a step to focus on and pause at.

### 4.6 Results & Future Work

Overall, I found from preliminary data and user interviews, that StepByStep improves the speed at which a visual task can be learned and that this improvement scales with the complexity of the task.

While this is a promising start, in the future we aim to evaluate StepByStep more in the context of a Pub and its relationship to the rest of the document. In particular, two key questions are yet to be understood:

1. **Engagement**

   Does the StepByStep interface cause more people to begin to learn a task that they would otherwise be unwilling to engage or learn with using simple linear
video? Once StepByStep is integrated into the public version of PubPub we may be able to study this by a/b testing pubs with one version using a StepByStep interface and the other using linear video. Engagement may be anonymously measured through time spent on the pub and clicks on the videos.

2. Clarity

For long or delicate processes, clarity not speed or engagement may be the main goal of a video interface. How can we measure whether StepByStep leads to clarity and a more precise execution of the process?
5. Video Comments

This chapter describes the approach, design guidelines and methodology for VideoComments, an interface that uses interactive video to create a novel method of Peer Review.

As described in detail in Section 2.5, Oral feedback is a key part of peer assessment literature, but is lacking in the scholarly peer review process. In this section, I explore how we can make it a part of the digital scholarly peer review process in PubPub.

While video is an excellent digital medium for audio-visual information, linear video functions poorly as a method of feedback because it lacks the ability to easily reference in elements of the document in question.

In this section I design and develop Video Comments, an interactive video system which pairs a user video recording with their manipulations on a document (scroll, select, click, highlight), in order to create a novel, rapid way of leaving audio-visual feedback on a document.

5.0 Goals

Video Comments has two main goals. First, is to create a viable commenting interface. This means that it must offer similar affordances to text commenting. It must be technically feasible, work across all devices, be simple to create and intuitive to navigate.

Second, I aim to explore whether Video Comments offers a meaningful alternative to text. This means evaluating two qualities of the comments: quality and type. Do Video Comments create comments of similar quality to text and if so, do they create new types of comments that are not possible using text?

5.1 Related Work
Video streaming as a method of teaching has become common place in online communities such as Twitch [39][40]. This method of stream of consciousness teaching allows knowledgeable streamers to reach a large audience with new knowledge without any production work.

In some cases, this may simply be a streamer going over a website or document and giving their thoughts and feedback[41]. However, this method of feedback and knowledge sharing currently only exists in the gaming and coding communities.

5.2 Design Principles

Based on background research and learnings from StepByStep, a few key design guidelines were identified before the development of Video Comments.

- Authoring Inspired By Video Chat

Video Comments should be designed in a way that to the author, is like they are speaking to their computer, inspired by interfaces like Skype and Google Hangouts. Capture of events and actions should be seamless and automatic.

- Coherence with PubPub

Video Comments backend design should be driven by needs for PubPub and scholarly publishing. The design guidelines for the data structure and storage of interactive video within PubPub is described in Section 3.

- Play Anywhere

Video Comments should be able to be played at any resolution or device and automatically adapt actions such that scrolling, selections and actions still work.

- Emphasize ‘Peer’ Assessment

Video Comments should should feel more like a friend showing you a document than watching a screen capture of them reading at a document.
Users should have agency over the direction of the document and be able to independently navigate and operate the document alongside the author’s commentary.

- **Navigable in Space & Time**

  While Video Comments can be navigated linearly through time, users should also be able to navigate it by section of comment, allowing users to skim the authors comments.

- **Default to Native Interactions Over Custom**

  Because users are not nearly as familiar with interactive video as with linear, defaulting to OS and device native behaviors is preferable. For example, use browser scrolling instead of hijacking elements to scroll.
5.3 Approach

5.3.1 Architecture

Similar to StepByStep, Video Comments is built entirely in Javascript using React.js for modular frontend code, Node.js for backend APIs and MongoDB for data storage and retrieval. Both the frontend and backend use NPM[38] for package management and dependencies. All of StepByStep is hosted in the cloud on Amazon AWS, uploaded video is stored in Amazon S3.

Unlike StepByStep, Video Comments is tied inherently to the PubPub document in order to listen to actions and so cannot be used outside the PubPub ecosystem without some adjustment.

The Video Comments AtomType is defined primarily by an uploaded video asset and a set of actions recorded by the user. Each action is timestamped and has some sort of spatial tag so that they can be access either linearly or through the document.

The schema for the VideoComments atom is shown below:
5.3.1.1 Scaling & Reflow

Video Comments must work across all devices, so we store ‘scaled’ positions. Scaled positions are relative to the overall document size and length and unrelated to the resolution at which it is viewed or created. For example coordinates (0.1, 0.5) indicate that the position is a tenth across the document and halfway down. As long as content does not reflow (change relative positions), this remains accurate.

This however creates an edge case in mobile rendering and mouseover actions. Mobile viewing is the only case that causes content to reflow[42], changing the relative positions of elements.

In this case, translation of mouseover events becomes lossy. The primary unit of HTML is a Node[43]. Nodes wrap around text, so for example in the following HTML code:

```html
<p>My first paragraph!</p>
```
Events will only fire on the 'p' tags, not individual words. We can only know which word it is over by looking at the x & y coordinates. However, if content refows (for example, ‘my’ and ‘first’ are on different lines) the x & y coordinates no longer have meaning. Thus, mouseover events on mobile viewing are disabled to avoid misleading actions. However all other events, because they target individual nodes, are enabled.

5.3.2 Interface

Video Comments is divided into two interfaces, the authoring and viewing interfaces. In both cases, the goal is to simulate real life oral peer assessment, for example, asking your desk mate to come over and read your paper while pointing out feedback.

Video Comments can be accessed at the following link:

https://pubpub-experiments.herokuapp.com/pub/VidVid-Moving_images_in_context

5.3.3 Authoring

The authoring interface of Video Comments is designed to mimic video chat interfaces such as Skype or Google Hangouts.
The user simply presses 'start recording' and their webcam is shown along with a pointer above their mouse. The following actions are recorded and timestamped: mouse movement, mouse clicks, document scrolling, element selection. After recording, the author can play their Video Comment before posting.

5.3.4 Viewing

The viewing interface for Video Comments allows readers to easily follow along with the author, while at the same time retaining control of the document. Users are able to pause, fast-forward and skip through the review. At any time, the user is able to interact with the document as they like. An example of the viewing interface is shown below.
Moving Images in Context

Andrew Lippman, Thariq Shihab

Abstract

Video has become so easy to make and distribute that it is becoming the most common way to describe many processes and ideas. However, it is often not effective at doing so. The video may be too fast, too slow, or it may demand too much time and investment on the part of the viewer. This thesis attempts to ameliorate this for a specific but useful circumstance: academic publishing. We do this in the context of PubMed, which is itself a radical revision of the traditional scholarly document. Instead of being a static, isolated checkpoint for knowledge, a PubMed presentation is a dynamic, collaborative, cross disciplinary forum for evolving ideas. Some of its main features are that it allows for the integrated inclusion of dynamic diagrams, raw research data, active programs, and linked text. In this thesis we develop techniques for embedding video illustrations into such publications. In particular, we develop techniques to subdivide the video into segments that link to closely related text, the means to easily perform that linkage, and models for exploration of embedded video documents. The result of this work will be a way to let the reader explore and contribute to the exposition of

An animated example of the Video Comment viewing interface. This is best viewed on the online link.

5.4 Results & Future Work

Overall, I found that Video Comments creates a viable method of commenting and review. Video Comments can be played across all devices and can be navigated and skimmed easily.

An evaluation for Video Comments was not performed in this thesis, but is planned for the future. During this test we will focus on evaluating the following:

1. Engagement

How do Video Comments affect the number of comments that are ‘review-like’? Are Video Comments more or less likely to be seen or replied to than text comments?

2. Quality

Are Video Comments of similar quality to textual comments? If not, how do they differ and are these differences useful?
3. Types of Comments

Does Video Comments lead to types of comments that are not impossible via text? For example, comments on diagrams that require pointing or navigating through interactive elements.

5.4.1 Possible User Study

A proposed user study is as follows. Two pubs of different content but a similar structure and breakdown of content (image placement and ratios relative to text, etc.) will be chosen. Users would then be asked to write a text comment for one and a Video Comment of another. Speed of commenting will be recorded as well as the users thoughts and feedback as they write it.

To compare contents, the Video Comments will be transcribed and compared to the text to see how they differ in the parts of the pub they address, i.e. content, style and structure. Any non-transcribable portions will be taken note of because they suggest capabilities of Video Comments beyond those of text.

Finally, users will be asked to complete a survey giving their feedback on Video Comments and comparing it to writing a text comment.

From this data we will analyze how Video Comments differs in terms of engagement, quality and types of comments.
6. Conclusion

This thesis describes StepByStep and VideoComments, tools that facilitate integration of video into scholarly documents for the purposes of instruction and review. This work included building an infrastructure within PubPub which enables their use in scholarly documents, coding and implementing both systems and evaluating StepByStep.

StepByStep and Video Comments are part of a broader movement in PubPub to modernize academic work. In this thesis, I have started to show that interactive video as part of this modernization can lead to more effective knowledge transfer and thus, more effective publishing.

In future work we plan to test these interfaces more broadly and open them for use by all writers in the PubPub ecosystem. By collecting usage data at a large scale, we hope to further illustrate the potential of using interactive video within scholarly documents.
References


[30] "Figshare". [https://figshare.com/]
[31] "Permanent links". [https://www.w3.org/Provider/Style/URI]
[39] "Twitch". [twitch.tv]
[41] Scarra. "Patch Rundown". [https://www.youtube.com/watch?v=kAt5YKhpors]
[42] "What is a reflow?". [http://stackoverflow.com/questions/27637184/what-is-dom-reflow]

63