

Introducing Smirk 2

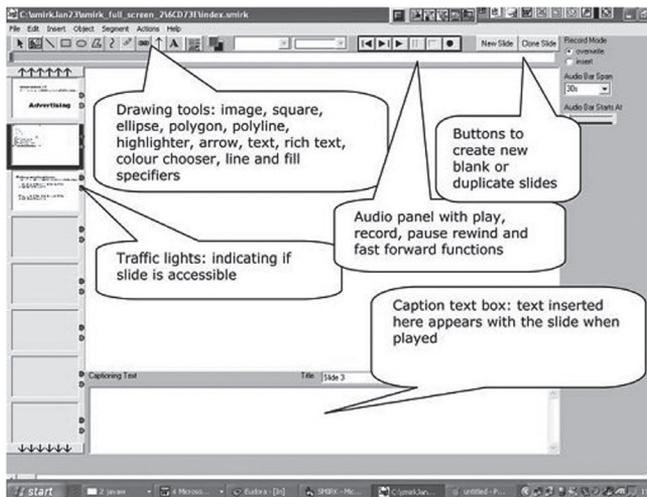
Steve Bennett, University of Hertfordshire
David Kraithman, University of Hertfordshire
Caroline Bettison, University of Hertfordshire

Abstract We introduce a new open source windows based lecture capture solution called Smirk 2. We describe the motivations for writing the tool, a comparison with other such programs on the market, and explain the mechanism of downloading Smirk on to a machine, either to use or to amend it. It also proposes potential projects that could be built on top of the Smirk platform – and also offers a justification of why academics should be involved in the creation not just the use, of learning technology.

Introducing Smirk 2

This paper introduces a new open source lecture capturing solution called Smirk 2 developed at the University of Hertfordshire. It builds on software developed by David Kraithman and Steve Bennett, Smirk 1 (Kraithman & Bennett, 2005). This development began in 2002 in order to fill a need then hardly catered for, namely to capture the content of a lecture. The reasons for this were manifold. At the time there was a lot of effort going into getting academics using PowerPoint. But this, we know from all kinds of anecdotal instances, tended to inhibit the creativity of lecturers and led to more wooden learning experiences. Also, once these items were uploaded to a Virtual Learning Environment (VLE), it seemed the lecturer's responsibilities were over and done with. And yet how limited a representation of what really goes on in a lecture theatre is the unaccompanied slides of a lecture? Smirk proposed to reunite the spoken narration with the slides in the VLE.

The original Smirk looked something like this:



It was different from most presentation authoring packages in a number of ways. Firstly the emphasis on accessibility: using Smirk there are constant reminders of the accessibility status of the presentation. At beside each slide thumbnail there is an indication of the accessibility status of the slide. And before every publishing (compressing the audio and video to make it viewable over the web) there is a dialog box informing the author of the accessibility status of the entire presentation: do all slides have titles? Are they all unique titles? Do all slides have captions (alternative text) for hearing impaired users? In addition, Smirk after publishing creates a page where any link or embedded media in the presentation is reproduced such that it can be perused in a non time-limited fashion (as might be the case if the hyperlink only existed for the duration of one slide).

Secondly, Smirk eschewed effects such as custom animations in favour of animation via timing and nothing else - something to be synchronised with the spoken narration - not as a means of mesmerising the spectators. This meant any dynamism in the presentation would be created spontaneously by the lecturer in live performance - rather than through a sequence of pre-cued flying bullet points. The third difference was that a Smirk presentation always regarded itself as comprising a multiplicity of media both in authoring and playback. Therefore, the presentation was not rendered down into a single unitary media (such as a video) and consequently any rendering it did do was much faster than for other authoring tools. Finally, Smirk was easy to use. Unlike PowerPoint, Smirk displays the audio panel all the time - making it very easy to record and amend audio. In PowerPoint conversely, the audio is hidden in the 'record narration' screen, involves some non-trivial setting up and is very difficult to edit after recording.

These concerns motivated the development of the tool, and by 2004 it was demonstrated for the first time to the University of Hertfordshire's annual Learning and Teaching conference to great acclaim. Later that year it went on to win the prize for teaching tools in the European Academic Software Award. It was also used extensively by the creators of the software in their own teaching. Steve Bennett obtained runner-up position in the THES E-Tutor of the year award that year when Smirk was used on a Computer Science Masters' course for students to make peer evaluations of other students' websites. David Kraithman successfully used it in a first year undergraduate Macroeconomics course teaching 870 students. Instead of a sequence of vast lectures of 200+ people, David created a set of online presentations supplemented by tutorials organised around them. Among the feedback received was:

"WOW, I am a 2nd year student and I failed this module. Why couldn't we have this help last year?? Basically I love it."

"I definitely agree that online tutorials are a good idea because without them I would probably not understand half this subject."

In 2005 the potential of Smirk was noticed by Capita Education Services who bundled it up in their 'Curriculum and Lesson Planner' tool, a product aimed at

the school market and teachers in particular. However, by the end of 2006 the software was beginning to look a bit long in the tooth, and moreover, a number of other well-financed tools had become available - and the need which Smirk was responding to in 2002 - was being catered for in different ways by many different market players. These included at the simple bundling-PowerPoint end, tools like Camtasia, Captivate, Impatica, all of which for a hefty whack could transform your recorded or narrated presentations into various forms of video or Flash. At the more video conferencing end there were tools like Elluminate or Adobe Connect - these could also create audio visual materials, but largely as a by-product of the lecture casting for which they were intended.

This forced the authors to reconsider the purpose and goals of the application. Since the market was now well populated and the resources we had were limited, it seemed hardly practical to attempt to compete on a commercial basis. However, the core values of Smirk (accessibility, immediacy, ease-of-use, speed-of-rendering) still seemed unsatisfied in many of the commercial offerings. Also other features such as online-quizzes were thought to be necessary - since a quiz seemed a perfect complement to a short online presentation. But at the same time the code really needed to be refactored - rendered more logical and modular. Finally there was the fact that other competitor software began to look better - they either used vector graphics or video - rather than Smirk's sequence of jpegs. Therefore the decision was made to totally revamp Smirk - as an open source community based project which would try to embed academic values in the tool we were creating.

The inevitable question made with regard to in-house learning technology is: why do it? Why not just leave it up to the software houses? The answer is that their values might not be ours. But there might be a more urgent ethical and philosophical question behind this: do we as academics deal with learning technology purely as an inevitable destiny which befalls us - or do we have agency in this process - can we influence the way it comes into the university? Can we have some input into the kinds of tools that we get enjoined to use?

Richard Sennett in his book "The Craftsman" talks about how craftspeople (and I would argue that teaching is indeed a craft) in the past have reacted to technology. Typically they faced threats from three sources: their employers, the unskilled labour sought to supplant them, and the machine itself. Sennett (2009) says craftsmen successfully defended themselves against employers and found common cause with unskilled labour, but were not able to fight the machine. He writes that they:

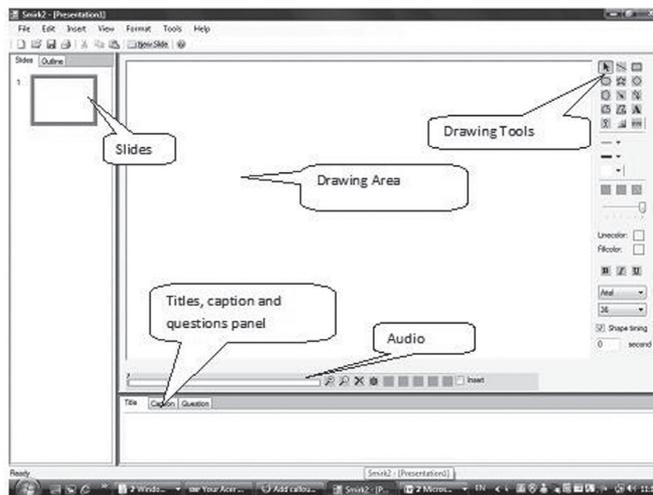
"...failed to invest in alternative strategies of mechanical design; the craftsmen did not sponsor research or themselves design machines that would keep a large body of skilled operatives necessary. Mechanical change came to the labour force rather than from within the labour movement. Failure on this third front has magnified the symbolic threat of the machine. Skilled operatives live with and through machines but rarely

create them in modern industry. Technological advance comes in this way to seem inseparable from domination by others.” – Richard Sennett (2009: 117-118)

Ultimately this was what Smirk 1.0 was about - creating lecture capturing software which still makes academics necessary - and preserves a model of academic practice where the academic is both the creator and the deliverer of that content. Tools coming from outside academia, and without significant input from teachers themselves, might significantly de-skill the work we do, or skew it in ways that make our work less rewarding to ourselves, and less valuable to our students.

These defining features of Smirk - live rather than pre-cued animation, swift rendering through media multiplicity, accessible outcomes - are the three core design features of new Smirk. However, in addition we have added significant enhancements. Firstly, we have added new output options. Instead of html and graphics being the primary output, now you can choose either Flash, Silverlight or Scalable Vector Graphics (SVG). This means that Smirk generates vector based accessible outcomes which look much better on screen. The second enhancement is the incorporation of multiple choice question technology. For each slide you can write a number of questions relevant to that slide.

The new Smirk looks like this:



As you see, many of the panels are in different and customisable positions, and there are new panels such as the question panel in which you can input simple text based question syntax which will probably cover 90% of the questions you might ever want to write. They can be multiple choice single answer, multiple choice multiple answer, fill in the blanks, ordering or association. Nonetheless, these questions can still output to Question & Test Interoperability (QTI) or Question Markup Language (QML) should you wish to incorporate Smirk

authored questions into server-based question banks. Similarly there are many more output options.

However, the most significant outcome is the open source nature of the new Smirk. The code of Smirk 2 can be downloaded from <http://code.Google.com/p/smirk2-1/>. This means that we now have a classroom capturing solution genuinely owned by academics.

Smirk 2.0 tries to follow the major guidelines for open source projects. It is intensely modular. The drawing panel, the audio recording panel, and the slide selection panel are all stand alone components which exist as separate sub-projects within the code structure. The software necessary for generating the various outputs such as Flash, Silverlight and html - this exists as a standalone exe - as does the software necessary for converting between various question formats. Other standalone exes include a Smirk 1 to Smirk 2 converter - as well as a presentation splitter. This enables large presentations (say hour long lectures) to be easily cut down into shorter mini-presentations better adapted for viewing on screen. Therefore while there remain a number of small rules and assumptions for programming Smirk, they remain fairly lightweight - but also allow most of the components in the structure to be developed independently of each other.

To begin programming and adapting the new Smirk all you need is Microsoft Visual Studio and a subversion client. Just point your subversion client to the Smirk repository at <http://smirk2-1.Googlecode.com/svn/trunk/>. This will download all the code necessary to run the application. Then open all the standalone exe projects one by one and compile them. And finally compile Smirk itself.

The Smirk repository contains a large issues log at <http://code.Google.com/p/smirk2-1/issues/list> and a wiki which will gradually be filled to explain the code and its structure. For the first release of the software, the major concentration of effort has been on creating a stable platform for lecture capturing. The next stage will involve the writing of add-ons and features which can take advantage of new and upcoming software and platforms.

However, Smirk 2.0 also exists as a simple program which you might just want to use - rather than code. To simply use the new Smirk - go to <http://smirk.herts.ac.uk> and go to the download page.

Therefore, if you have some interesting idea you might wish to try out in a lecture - with a little tweaking of Smirk 2.0 you can very quickly make it a reality. For instance, suppose you wanted to display a Twitter feed that your students could tweet to during a lecture whose content would be projected on screen and be saved there after? All you would need to do would be to write a visual basic component which could consume a Twitter feed, and then convert the tweets into text objects which could appear on the screen. An idea I have would be to run a lecture where all my students collaborate in scoring other students'

work (probably via mobile phone) - but I would like to give extra marks to those students whose evaluations were nearest to mine. That would need to be done by calculating the co-variance scores between each student's pattern of marking and my own - and giving extra marks for the students with the lowest covariance scores. For effect this would need to be projected on screen. This would require someone to write a small mobile application which sends scores to a web page - a web server program which collates those scores. Finally what about speech recognition going on in the background and being broadcast to students' handheld devices via Twitter/Google talk/MSN?

These are the kinds of projects which with only modest funding could be attempted in that the basic infrastructure will have already been created in the Smirk 2.0 platform. The software is also written in one of the easiest languages of all Visual Basic - which means in a negative sense, that the software is restricted to the windows platform, but in a more positive sense, requires very little programming expertise in order purposefully amend.

Smirk 2 has been designed and written by Steve Bennett, David Kraithman, Saisakul Chernbumroong and Caroline Bettison. Since we won't get any money from it, our only reward is that people use it and contribute code to it. Therefore we commend it to you and look forward to seeing your presentations and amendments!

References

Kraithman, D. & Bennett, S. (2005) Blending Chalk, Talk and Accessibility in an Introductory Economics Module. [Online] Available at: http://www.economicsnetwork.ac.uk/showcase/kraithman_smirk.htm [Accessed: 6 January, 2005]

Sennett, R. (2009) *The Craftsman*. Penguin