Using Zoom Technologies To Display HEP Plots and Talks

Paper Website

Gordon Watts

University of Washington, Seattle

Abstract

Particle physics conferences and experiments generate a huge number of plots and presentations. It is impossible to keep up. A typical conference (like CHEP) will have 100's of plots. A single analysis result from a major experiment will have almost 50 plots. Scanning a conference or sorting out what plots are new is almost a full time job. The advent of multi-core computing and advanced video cards means that we have more processor power available for visualization than any time in the past. This poster describes two related projects that take advantage of this to solve the viewing problem. The first, Collider Plots, has a backend that looks fro new plots released by ATLAS, CMS, CDF, and DZERO and organizes them by date, by experiment, and by subgroup for easy viewing and sorting. It maintains links back to associated conference notes and web pages with full result information. The second project, Deep Conference, renders all the slides as a single large zoomable picture. In both cases, much like a web mapping program, details are revealed as you zoom in. In the case of Collider Plots the plots are stacked as histograms to give visual clues for the most recent updates and activity have occurred. Standard plug-in software for a browser allows a user to zoom in on a portion of the conference that looks interesting. As the user zooms further more and more details become visible, allowing the user to make a quick and cheep decision on whether to spend more time on a particular talk or series of plots. Both projects are available at http://deeptalk.phys.washington.edu. The poster discusses the implementation and use as well as cross platform performance and possible future directions.

Three Tools

Three tools have grown out of the original vision. I can't travel to as many conferences as I'd like and I was looking for a way to quickly browse conferences. Something that would allow me to quickly look at a conference over a lunch break, examine a talk or two, quickly load a talk, glance at a few slides, etc. The current state-of-the-art is pretty good—and indico agenda, PDF's or PPT's. But that interface isn't built for quick glances—it is built for deep and careful examination. Or presentations.

I adopted a zoom interface (ala a mapping program like google maps). This was presented in CHEP 2009. The tool has continued to evolve since then. Simple evolution: more talk formats are supported (text documents, OpenOffice documents, etc.). But it has also bifurcated into three different tools.

DeepTalk — this is the original project, and can be seen at http://deeptalk.phys.washington.edu. An Indico agenda is downloaded and layed out in a zoomable grid-like pattern allowing easy exploration with a browser with Silverlight, a good network connection, and a mouse wheel (or touch interface that allows easy zooming). Source code is at http://deeptalks.codeplex.com/.

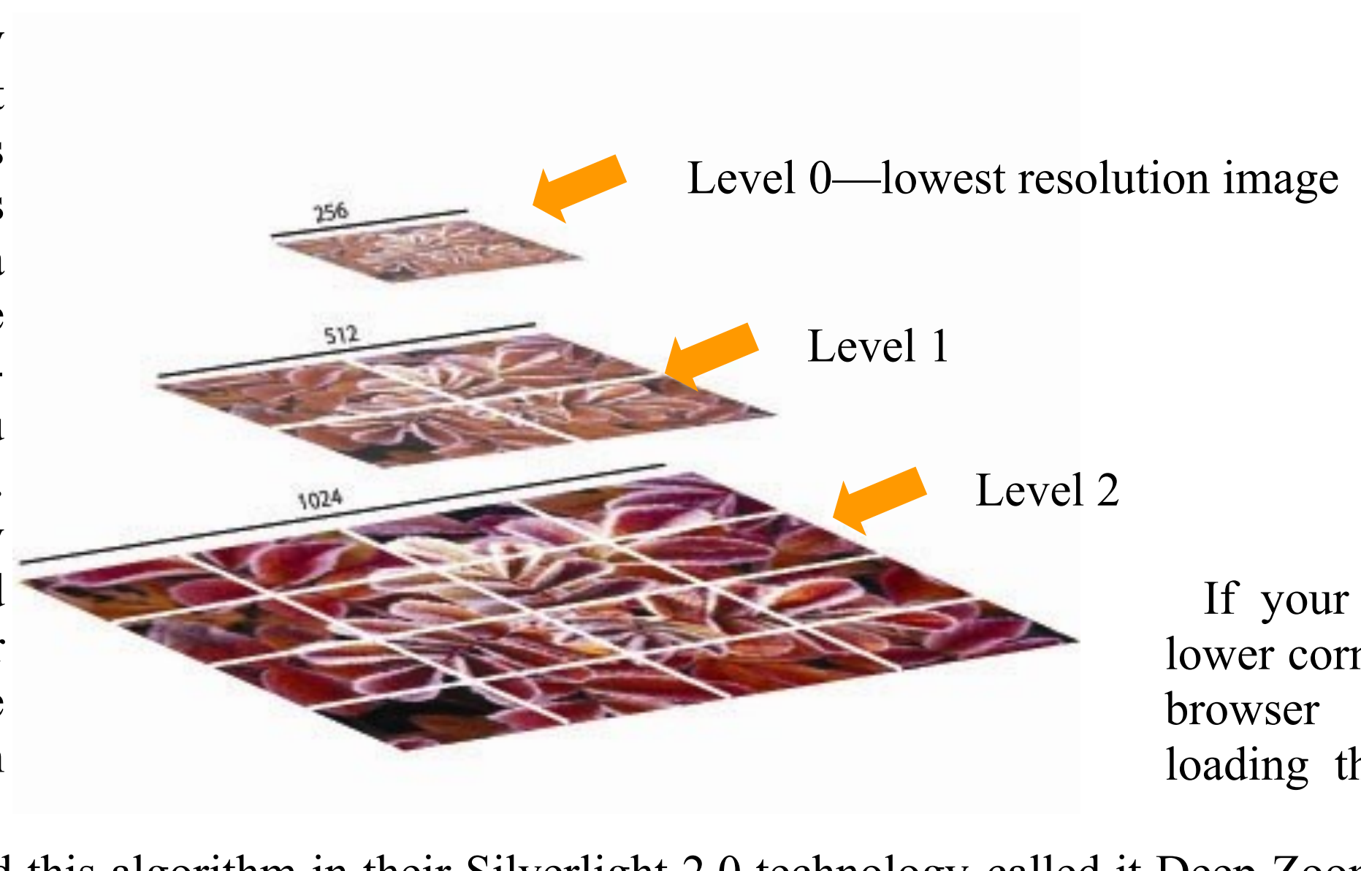
DeepTalk Personal — This is a windows application that allows you to deep zoom a single file (like a talk) or a directory, or a publically accessible Indico agenda. It includes the ability to examine meta-data in a PowerPoint pptx file and add additional documents and items. Source code and installation is at http://deeptalks.codeplex.com/.

Collider Plots — A backend that scans ATLAS, CMS, DØ, and CDF for new plots in their public plot page and catalogs them along with meta-data about the plot. And a Silverlight based interface that allows you to sort them by various criteria like release date, topic, or search the captions. Source code can be found at http://particleplotpivot.codeplex.com/.

All these tools are open source, and written in C#. You can find the source code at the open source codeplex site. There are common libraries I've written that are shared between these projects that are also hosted on codeplex.

What Is Deep Zoom?

The algorithm is fairly common, actually. Most web mapping programs use it. The iPhone uses it. As you zoom in on a map, the map web site loads more detailed images of just the area you are interested in. Zoomed out you may only see major roads and land features. The closer you get the more you see of city streets or even houses.



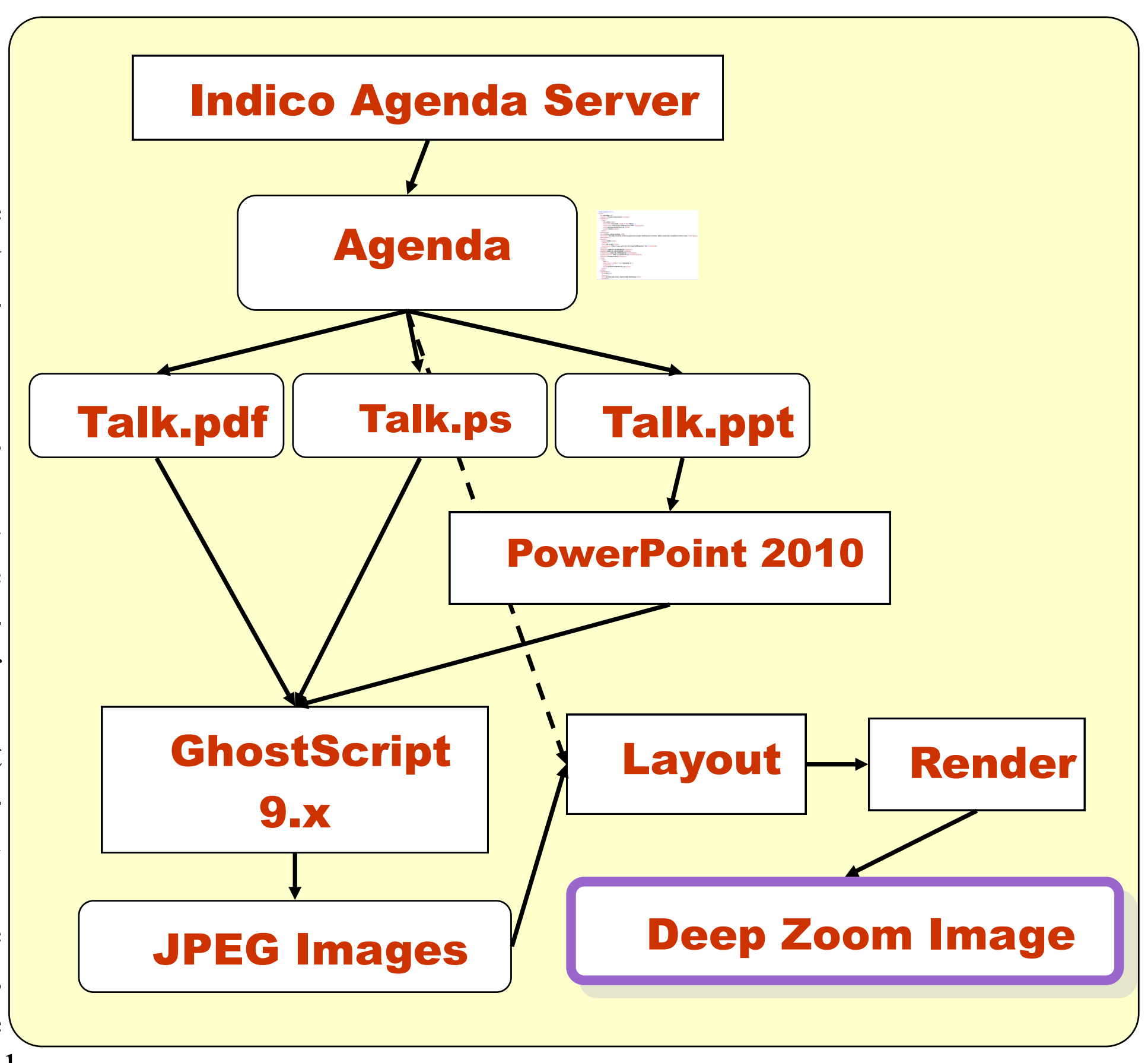
If your view is only this lower corner, why should the browser waste bandwidth loading the rest of the images.

Microsoft implemented this algorithm in their Silverlight 2.0 technology-called it Deep Zoom. The high resolution image is rendered as a single low resolution image, 4 higher resolution images, and then 16 very high resolution images. The web browser is instructed to load only the relevant images. Zoomed out it will load only the Level 0 image, and as the user zooms in it will select the appropriate images from Level 1 and Level 2.

Instead of the picture of the flower, think of all the slides of a conference laid out on a big floor. That is the high resolution image. A decent length talk rendered this way can require almost 20 levels when rendered at 200 DPI. Each tile is configured to be 256x256 pixels. When the jpeg is written with a 95% quality level the sizes vary between 4 and 30 KB each, depending on complexity—small files that can be quickly downloaded over the internet.

Once the images arrive at the web browser, Silverlight stitches them back together and scales them correctly to appear in the user's web browser.

Rendering



All three tools use the same or very similar code to render the images into the DeepZoom format. There are two key components that are being used - everything else could be replaced with little effort.

- *Silverlight 4.0*. This is Microsoft's competitor to Adobe's Flash/AIR. The key thing is the *MultiscaleImage* control. This control does the heavy lifting on the user's computer: it blends images as the user zooms in, it coordinates the http requests for the base images, and it stitches the tiles together. Silverlight works on Windows and Mac OS X and in most modern browsers like IE, Firefox, and Chrome (not Opera). Linux is coming as Novel gets their open source *Moonlight* project off the ground. There is javascript code that will emulate a limited set of the features used by the control, but significant work is required to complete the conversion to a non-Silverlight based viewing experience.
- *Composer Tools*. This is an application and a library that can be used to generate the Deep Zoom images. As input it takes a data structure that describes the layout of a collection of images, and the image files themselves. These tools have been implemented by many open-source projects now—converting away from the proprietary would probably be easy, and might improve efficiency as they get away from the single-threaded requirement.

Silverlight is the only external install used on the user's computer. On the render computer a range of software is required to turn an agenda into a Deep Zoom image. There is no reason this toolset couldn't be expanded in the future:

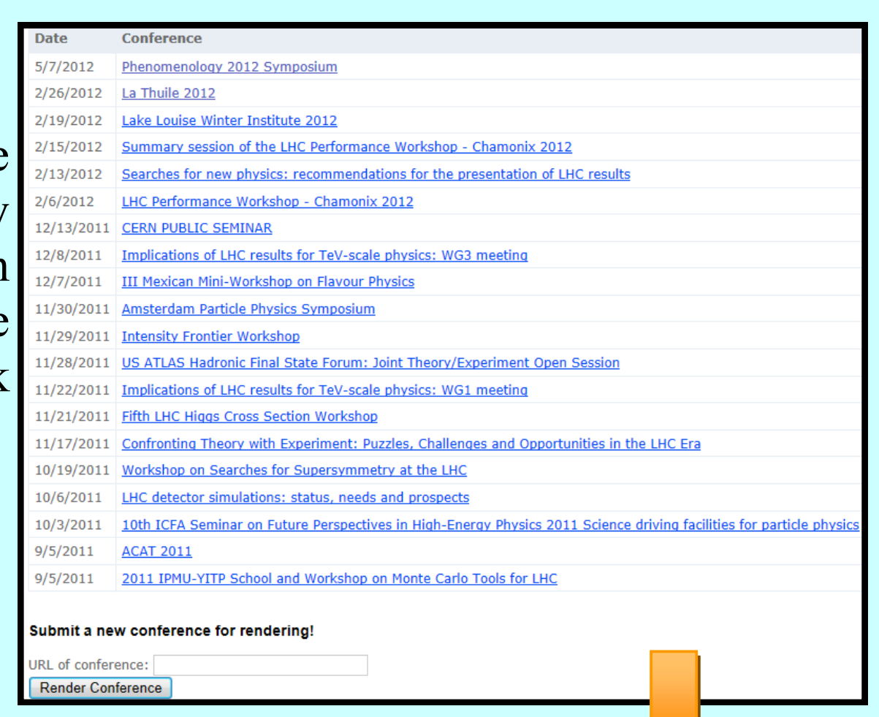
- Custom code downloads and parses an indico agenda in XML format or scans files on the local computer or remote http pages for plots. The information in is free form—for example it is hard to get all talks but avoid papers. Worse, MIME-Types are only set for some sub-set of document types, and often without those these tools can't tell what format they are looking at (they make a best-effort to guess).
- *PowerPoint* is used to turn a ppt file into a PDF file. I do see a few crashes with PPT that contain unknown picture formats (mostly PPT generated on Apple computers).
- The *GhostScript(GS)* program has the job of turning a single talk (pdf or ps format) into a sequence of jpeg images. There are lots of problems with GS. GS crashes on lots of PDF files that Adobe Acrobat renders correctly. It doesn't always get talk rotation correct; I've got code to fix up most cases and re-rotate the image. However, it isn't perfect. GS has trouble with funny DPI settings (e.g. output from the TeX slide package). Sometimes adds oceans of whitespace around the outside of the talk content. As a result of this I no longer think of PDF as a standard. :-)
- There is no reason other formats can't be rendered. The architecture is completely plug-in based, and a new format is just a matter of writing a new plug-in.
- Layout is done by some custom code—a simple set of in-memory transforms. No images are loaded to accomplish this, so this is quite memory efficient. The in-memory meta-data can contain hot-spots that alter navigation or open an external link in a browser.
- Rendering is done by the *Composer Tools* referenced above. Each talk is rendered separately. Agendas are living things: the code is designed to only re-render things that have changed, as rendering is very expensive. A single talk can take 20 minutes or more, depending on the number of slides. Rendering is also a memory hog— a 50 page talk can take more than 4 gigs of memory.

Finally the server is built out of several off-the-shelf server products. There is no reason this couldn't have been served from a Linux box using PHP, for example (or any other framework or platform). A very nice thing about Sliverlight is that it can be distributed as static content from any server platform. Indeed, once the talk is rendered this is very much just that—static content. For the personal DeepTalk for local rendering a small web server is included in the app. However, just a whole-sale copy of the rendered data to any web server will work just fine (as long as the MIME type is setup for serving Silverlight).

DeepTalk
http://deeptalk.phys.washington.edu

Jump Off Point

The address http://deeptalk.phys.washington.edu lists the most recent 20 conferences and you can enter the URL of any conference you wish to view. Clicking on a link or entering an already known URL will take you to the initial view of the whole conference. If you enter an unknown URL, DeepTalk will queue the rendering request.



Date	Conference
5/7/2012	Phenomenology 2012 Symposium
2/26/2012	La Thuile 2012
2/19/2012	Lake Louise Winter Institute 2012
2/15/2012	Summary session of the LHC Performance Workshop - Chamonix 2012
2/13/2012	Searches for new physics: recommendations for the presentation of LHC results
2/6/2012	LHC Performance Workshop - Chamonix 2012
12/13/2011	CERN PUBLIC SEMINAR
12/8/2011	Implications of LHC results for TeV-scale physics: WG3 meeting
12/7/2011	III Mexican Mini-Workshop on Flavour Physics
11/30/2011	Amsterdam Particle Physics Symposium
11/29/2011	Intensity Frontier Workshop
11/28/2011	US ATLAS Hadronic Final State Forum: Joint Theory/Experiment Open Session
11/22/2011	Implications of LHC results for TeV-scale physics: WG1 meeting
11/21/2011	Fifth LHC Higgs Cross Section Workshop
11/17/2011	Confronting Theory with Experiment: Puzzles, Challenges and Opportunities in the LHC Era
10/19/2011	Workshop on Searches for Supersymmetry at the LHC
10/6/2011	LHC detector simulations: status, needs and prospects
10/3/2011	10th ICFA Seminar on Future Perspectives in High-Energy Physics 2011 Science driving facilities for particle physics
9/5/2011	ACAT 2011
9/5/2011	2011 IPMU-YITP School and Workshop on Monte Carlo Tools for LHC

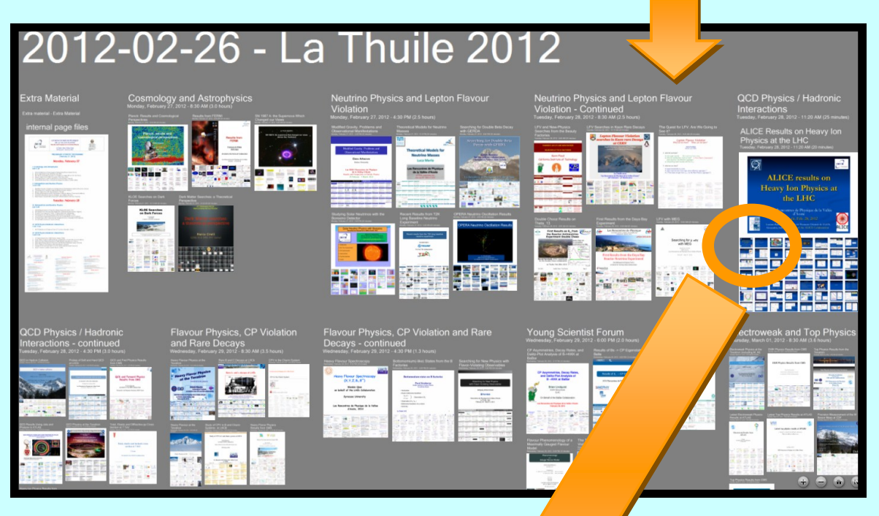
Submit a new conference for rendering!

URL of conference:

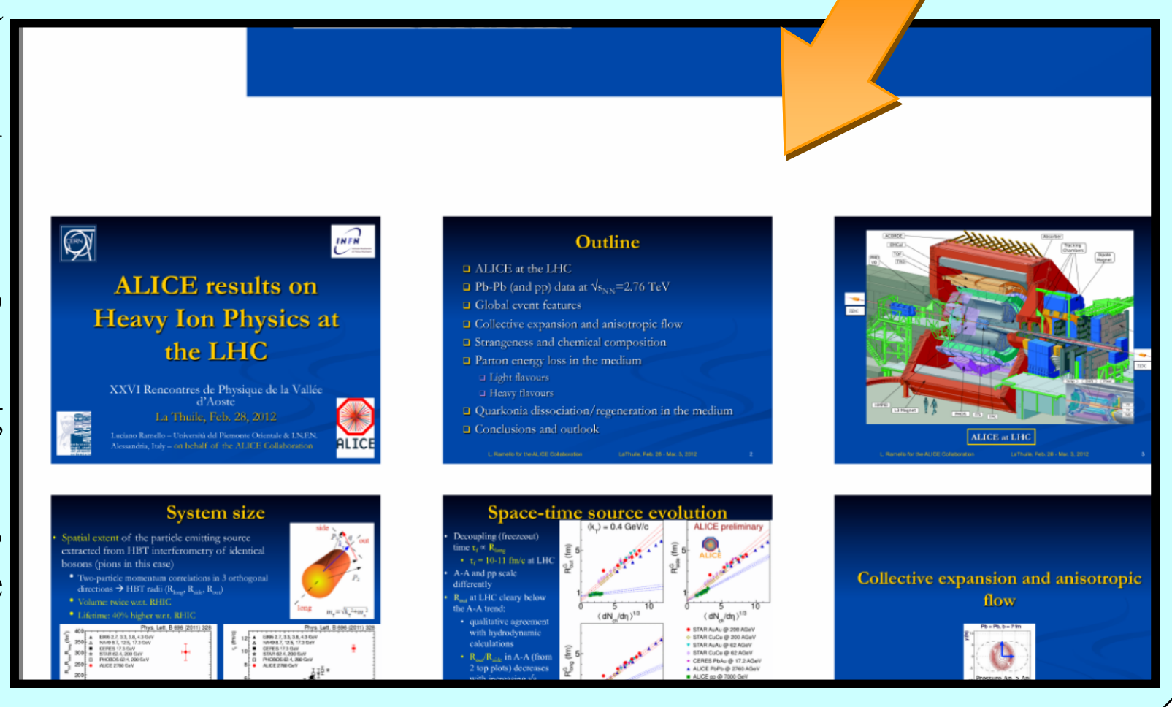
Conference View

The complete conference is rendered as a single picture. You can zoom in or out using the mouse scroll wheel (or clicking on the image or using the "+" and "-" buttons). Panning is accomplished by click-and-drag with the mouse or the arrow keys.

It is possible to quickly zoom in, out, and pan around and explore all the talks in the conference. The layout should make it easy to explore what you want to look at quickly, allowing your eye to determine if you want to delve deeper into a talk.



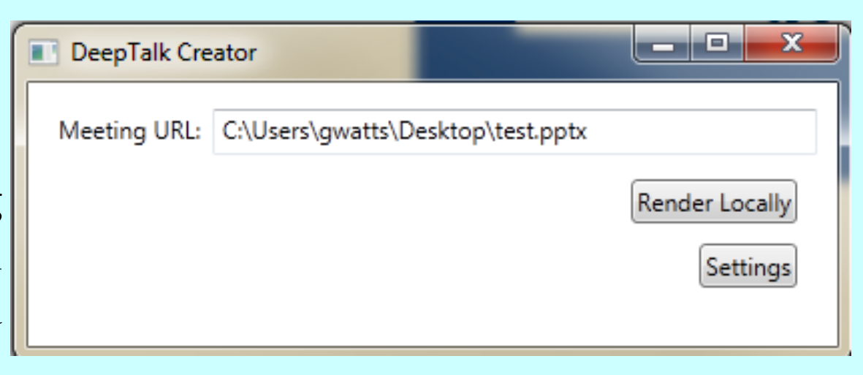
Below is a quick example of zooming into a conference.


ALICE results on Heavy Ion Physics at the LHC
Outline
System size
Space-time source evolution
Collective expansion and anisotropic flow

I have found the following makes this tool most useful to me:

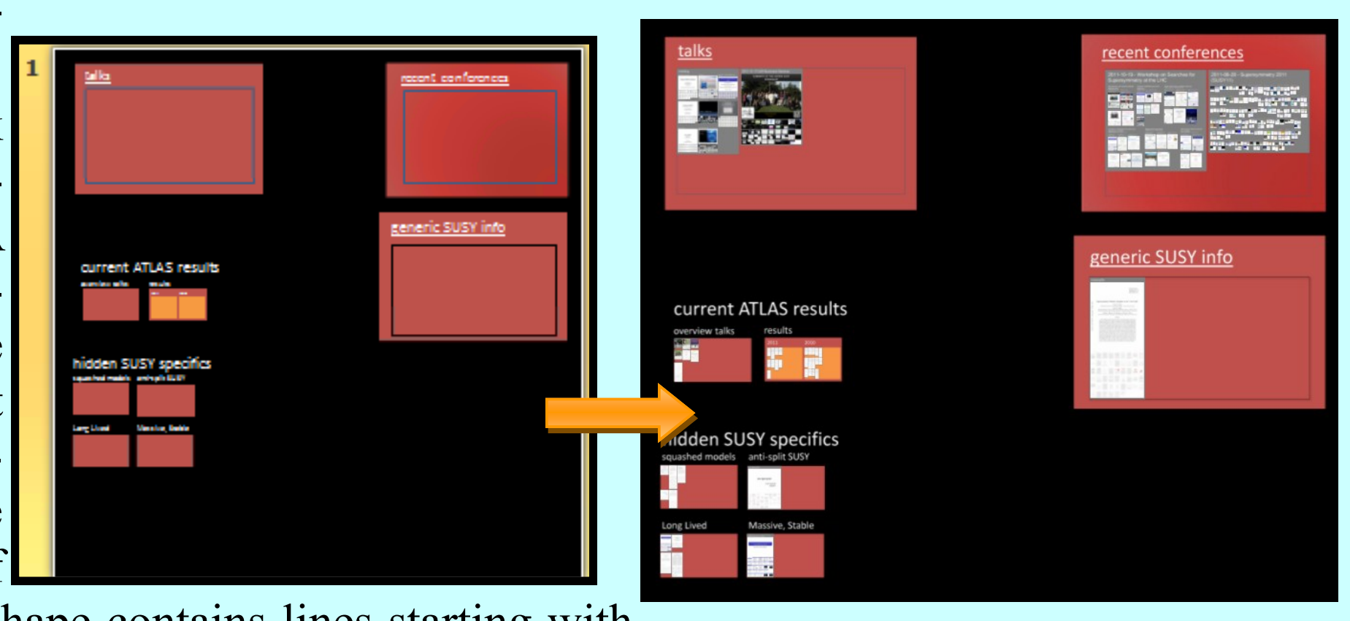
- Make browser full screen
- Turn off status bar at bottom of browser so you have nothing-but-talk on the screen.
- Use a mouse with a scroll wheel—zooming in and out is much faster!
- Use a high speed internet connection. Lots of small jpeg's have to be moved from the server to your browser!

Personal DeepTalk
http://deeptalks.codeplex.com



Meeting URL: C:\Users\gwatts\Desktop\test.pptx

This tool is very similar to DeepTalk described above. Indeed, it uses the same code to render. However, instead of starting from an Indico agenda from a database, it lets you enter a path or http address or Indico agenda and starts from there. You could do a private rendering of a conference if you wished.

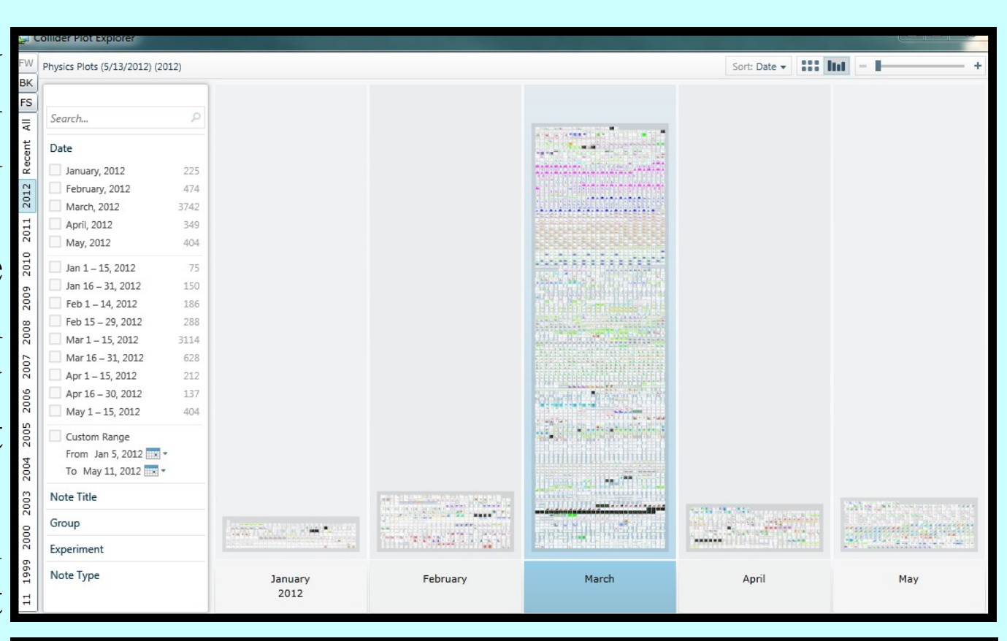


It has recently been extended to look inside a pptx file and alter its rendering depending upon what it finds. A pptx file is just a zipped file containing XML describing the layout of the PowerPoint presentation. The format is standardized, and there are a number of libraries that will parse it. The slide on the left contains a number of empty rectangles. If *Alt Text* for the shape contains lines starting with *render:* then the rest of the line will be treated as a resource to be rendered, and placed inside the rectangle. For example, if *render:2011-11-17-UW-Summary-Seminar.pptx* is in the rectangle, then that PowerPoint slide deck will be rendered. You can see this on the left, which is how the result appears in a web browser after the original pptx has been rendered.

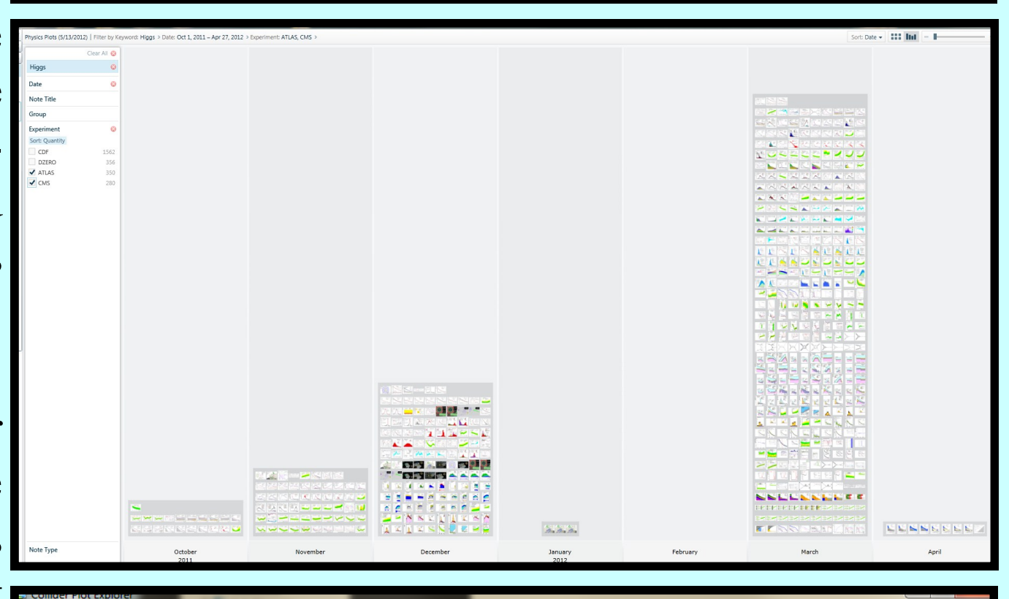
I've used this to help prepare for conferences, but I suspect there are other uses for it: I'm still exploring and learning.

Collider Plot Explorer
http://deeptalk.phys.washington.edu

This tool attempts to track new plots as they are released by the experiments. Its motivation was simple: I can barely keep up with new results coming out of my own collaboration, let alone all the others.



There are two major components: the back-end and the viewer. The viewer is shown at left. It is, again, based on Silverlight. In the view shown the plots are organized by date, each small square is a plot. The pane on the right side allows one to further window the data displayed. The data is most often displayed by year, as shown in the strip on the left, but this has more to do with efficient loading of data than anything else. Even from this range you can tell quite a bit of information. For example, here is what this looks like for plots from the LHC that mention the word *Higgs* from October 2011 to today. You can see the December seminar release of the early Higgs results, and then the "Morriond bump."



A great deal of metadata is stored for each slide as well. Plots can be zoomed in on by either clicking on them or by zooming with the mouse wheel, as shown third in the series at left. The figure caption, links to the graphics files for the plot (eps and png in this case), the date it was first seen by the web crawler, a link to the web page where it was found, what experiment, and what group are also included. All of these fields are search able and most are sortable (and you can histogram in them as well).

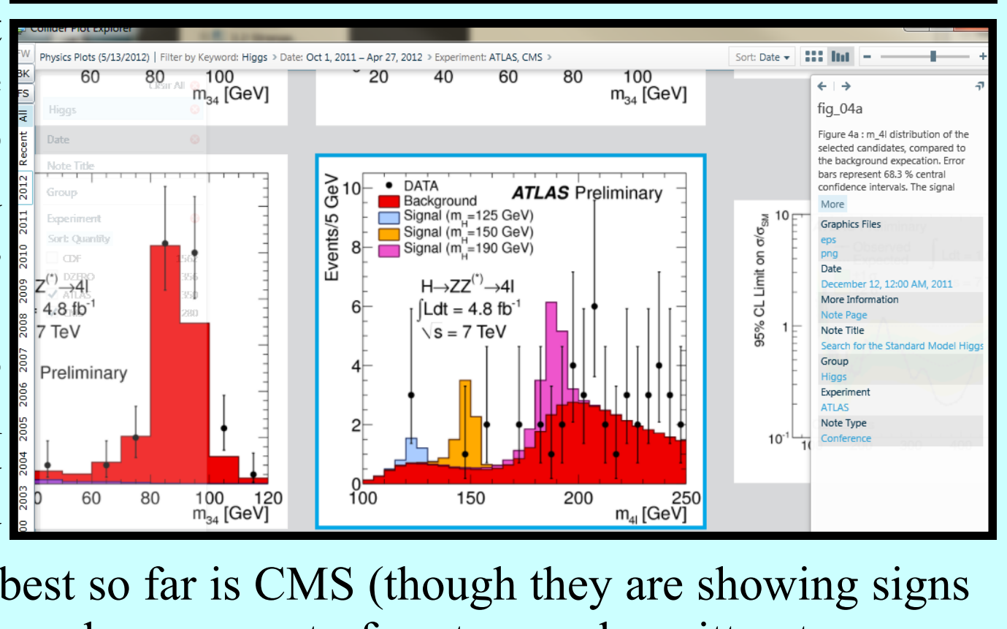


Figure 4a : m_4l distribution of the selected candidates, compared to the background expecation. Error bars represent 68.3 % central confidence intervals. The signal

All the metadata must be scrapped off web pages. No experiment uses a standard format. Not only does each experiment present its pages in different formats, they are often not consistent between experiments. CDF is by far the worst, DZERO next worst, then ATLAS, and the best so far is CMS (though they are showing signs of devolving with their new twiki pages). There is a tremendous amount of custom code written to scrape the information off the web pages, and when an experiment changes the page format my scrapper often breaks. I do my best, but I don't completely keep up.

Another big missing source of plots are published papers. Only plots on experiment pages are grabbed, and any multi-page PDF file is, in fact, ignored. This includes anything that is pointed to on the archive. A possible upgrade path is inclusion of those plots from experiments that don't also post them in their standard format.

Finally, this code needs improvement to be more gentle to the web sites is scans. It runs once a night, and there are some files it downloads repeatedly because it fails to render them and has no memory of its past failures.

This tool gets its most heavy use right before I have to give a talk at a conference, as you might expect. I often look once or twice a week to see what has come out from the other experiments as well – just to see what I've missed on Facebook...

Resources and Performance

Conclusions

The original system has evolved into three projects. For me personally, all three get low-duty cycle usage. I find all three useful at different times, and my usage patterns vary from intense usage to little or no usage. Development also proceeds in similar fits and starts, depending on what I want or need. On the other hand most of this code is quite robust. It is often months between me having to work on the systems due to a problem.

There are plenty of things that need work. This system is a resource hog—it requires high bandwidth network, lots of memory, and lots of disk space. Most of the tools worked well for the project, though some of them are more complex than they need to be for this project (i.e. the web service infrastructure for the DeepTalk web site). It was very nice to be able to program end-to-end in a real programming language (C#) - from the server (CLR) to the web browser (Silverlight) - and without the tools to render the multiscale images I doubt I would have attempted this project. In the end a small machine serves data for the plot explorer and the DeepTalk browser. It can do this in part because both of these require only static content—no web app runs on the server except to serve up the list of talks and control the back end rendering for DeepTalk.

The rendering runs on my desktop machine, a 2 year old mid-range i7 with 12 GB of RAM. This machine labors under this work and other work I have assigned to it (the build and test server runs on this machine as well). In particular for this project a number of optimizations could be done to reduce processing time. For example, remembering from night-to-night that a particular download was not valid.

DeepTalk has a lot of information available—it would be nice to be able to search all the talks similar to the way plots are searched in the Collider Plot Explorer. Perhaps even combining the two. A search interface that would speed talk exploration from the top level would also be very nice—perhaps find all the talks by a particular person. Finally, auto-discovery of most conferences would also been a boon. There are a number of well known Indico sites (like CERN) that host conferences and scanning that site would be a first step.

A more major, and necessary, but not interested upgrade, will be moving the display backend away from Silverlight and towards JavaScript. However, that requires work in an area I have very little knowledge: browser programming with JavaScript.

Finally, Indico has recently made a list of updates—including providing an API. A very nice feature of this API is that once an use a key to access otherwise protected events. This will allow one to render ATLAS or CMS meetings with this tool—something not possible up to now due to the single-sign-on requirement at CERN for protected meetings.

This API also means a number of other ideas are possible for future development. No need to use the browser to access Indico meetings any longer—once an write an app!

Modern computing infrastructure is much more powerful than 10 years ago. However, our basic method of presenting results to ourselves and the scientific community hasn't evolved very much. All the above tools attempt to harness the GPU, for example, and to collate talks in ways not done before. But it feels like so much more can be done and I don't think I've hit *the solution* yet.