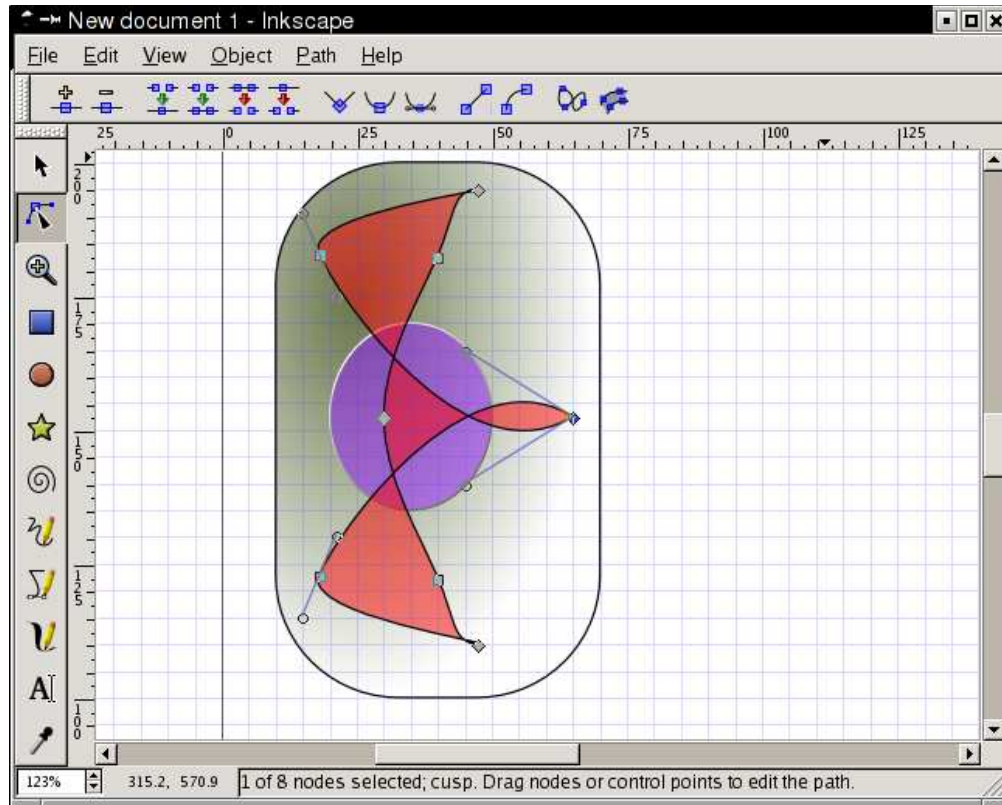


Inkscape

A Union of Contributions Makes a Difference

Bryce Harrington



I had to reread Bulia's email three times. It was one of those nigh-mythical events that you read happens in Open Source projects, but never see in person. Yet, amazingly, here it was, and I knew Fred's patch would instantly double Inkscape's utility.

Scalable Vector Graphics and the Open Source Community

Inkscape is a program for viewing, making, and editing two-dimensional vector drawings. This is different from "raster" drawing (figure 1), as in MS Paint, Photoshop, or The GIMP. In those tools you're essentially just "painting" destructively on a canvas. By "vector" (figure 2) drawing I mean that when you create a shape like a rectangle, it retains its identity. You can easily go back and resize it, change its color, or move it around without disturbing the rest of the drawing. Vector drawing is what you'd be doing in Illustrator, Corel Draw, Freehand, Dia, Visio or even PowerPoint (eww...).



There have been a number of popular Open Source vector graphics tools such as *tgif*, *idraw*, *Sketch*, and *xfig*, but one of Inkscape's distinguishing features is that it stores its drawings in a web-friendly XML format – SVG. SVG, an acronym for "Scalable Vector Graphics", is a W3C standard that is gaining support worldwide, in proprietary and public software alike.

The Open Source community is now adopting the SVG format for everything from desktop icons and company logos to web page animation and artistic Illustration. Inkscape (by way of Lauris Kaplinski's popular *Sodipodi* project) is derived from *Gill*, one of the first Open Source SVG editors, and so follows a long history of serving the SVG needs of the community.

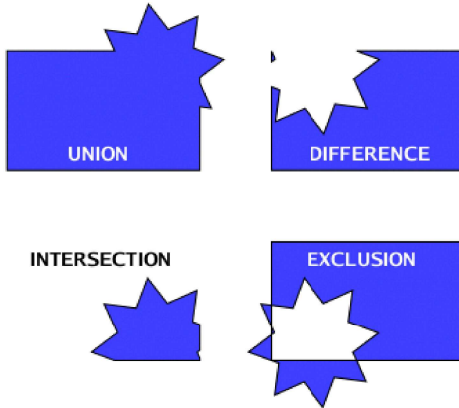
In the five years since Raph Levien began work on *Gill*, a huge range of features and capabilities had been added to the codebase. Node editing, alpha blended gradients, object alignment, text handling, localization and more had augmented the basic underlying drawing capabilities to make the tool potentially useful for real drawing work. However, there was one glaring omission for which we and scores of users had been seeking a remedy...

The Contribution of Boolean Operations to Inkscape

I read the email again just to be sure.

"I've been sent a new patch that implements boolean operations... The – license is public domain.
It's been uploaded to the patch tracker." – Bulia, November 2003

This was very cool. Boolean operations are a way of taking two shapes and combining them together in various ways to create a single resultant shape. Users of Adobe Illustrator might recognize them in the "Pathfinder". The four basic operations are Union, Difference, Intersection, and Exclusion. It's an absolute requirement for creating any artistically sophisticated drawing, and it's lack had held the tool back.



Once before, someone had contributed a patch to add boolean operations, but that patch relied on a polygon clipping library provided under an incompatible license. There's little more frustrating than having a solution in hand, only to be hamstrung by legal problems. Even though it was an important feature for us, we regretfully postponed development of it into the distant future on our roadmap and proceeded with other work.

Here in my inbox, **unsolicited and totally unexpected**, was the answer. We quickly double-checked that the licensing was clean, that the code was the author's original work, and that it indeed implemented the feature as promised; it passed on all counts. Fred's boolean patch had arrived right as we were releasing Inkscape 0.36, so as soon as that release was out the door we merged his patch and started working with it.

Following our policy to "Patch first, ask questions later", we integrated the new feature as soon as practical, without wasting time arguing about it on a mailing list. We figure that the best way to evaluate an idea is to code it up and see how it works in practice. A working feature now is better than a perfect implementation that still isn't done. Along with that, maintaining a low barrier to entry for new developers is vital; we don't want anyone to give up on contributing out of fear their contributions won't be accepted.

One of the first areas of focus was to add menu items and keyboard shortcuts for the commands it provides. Mentalguy had just recently finished up a massive redesign of the user interface, and so knew the exact place to slip it in. Likewise, Bulia had been hard at work creating a thorough set of key mappings for giving experienced users fast shortcuts for productivity, and found appropriate keys to map the new commands to. Mentalguy's attention to menus, coupled with Bulia's shortcut key work demonstrate the value placed on usability for this tool. Both men are artists at heart, so place great importance on making the tool as useful to real users as they can.

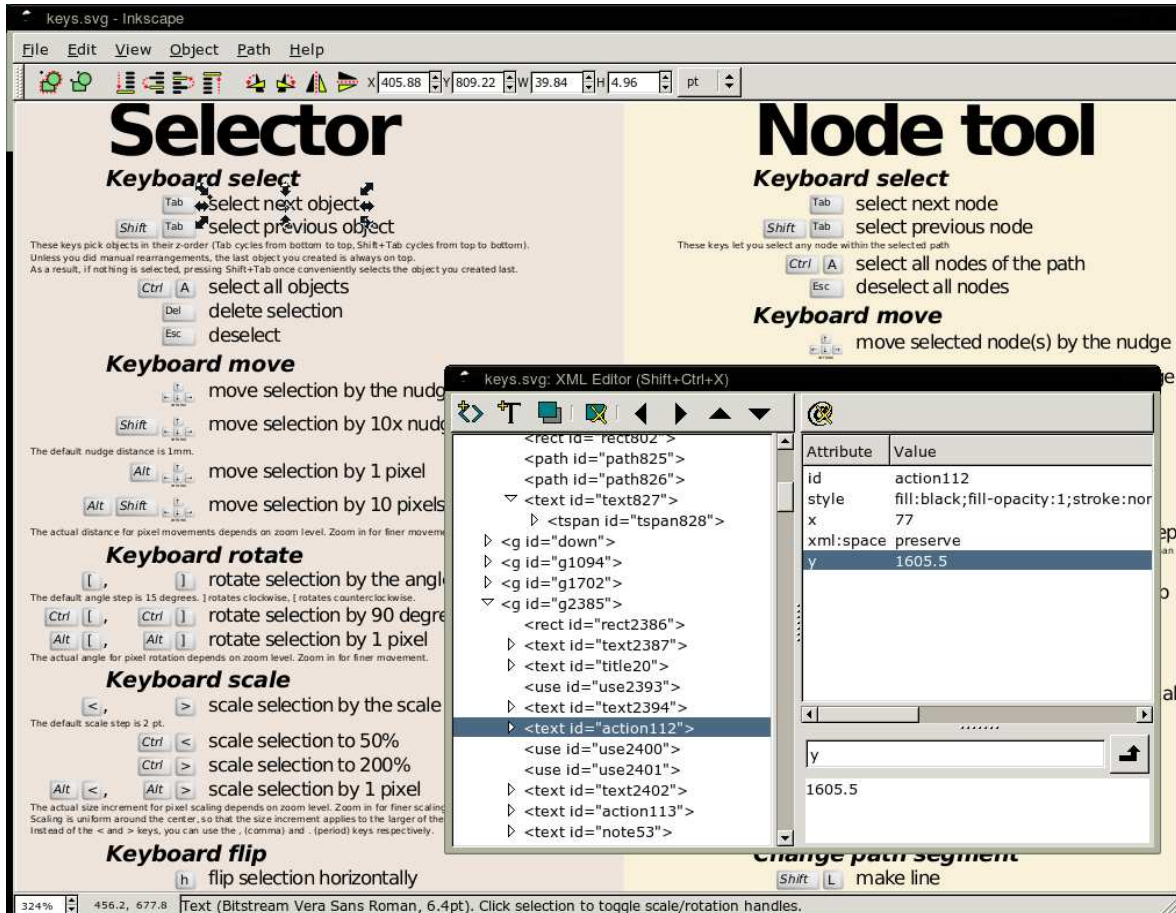
Once the basic functionality and usability of the newly merged booleans code was established, other users started testing the feature in real world situations, and identify and shake out bugs with us.

Meanwhile, the development team turned their own focus deep into the code itself. With Fred's help, they isolated and repaired lingering quirks and bugs. Our more mathematically inclined developers dug into the algorithms in search of ways to relate them to similar functionality elsewhere in Inkscape. By the time we released 0.37, the booleans code was solid.

Even better, Fred stayed on and worked with us to optimize the code's performance and derive new and exciting features on the foundation he built – such as commands for offsetting, division, and simplification of paths.

In many Open Source projects, such a windfall would be rare, but Inkscape has been fortunate to receive many major advancements from the contributions of newcomers. This is evident when one has used the program for a bit and notices that beyond just having all the usual drawing capabilities, there's a huge number of thoughtful-yet-modest behaviors hidden away, that combine to make Inkscape a pleasure to work in. Grid and guideline snapping uses a tunable "gravity" style rather than absolute snapping. Objects can be nudged a pixel at a time, for

when you need to get that shape just right. The zooming and auto-scrolling of the canvas are direct results of feedback from users and their ideas of how things "should" work. There is even an XML editor built in, for those power users who like to see exactly what they're getting; this tool allows direct access to the drawing's underlying DOM model. SVG is XML, and Inkscape is not ashamed of that fact.



The Future

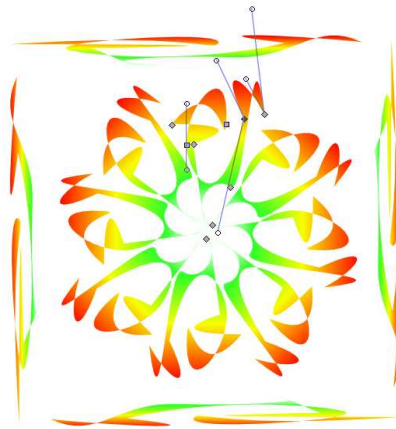
Work continues strong on Inkscape. A very interesting new feature that will appear in the 0.39 release¹ is *clones*. Basically, you can create a copy of a given object that inherits its properties, so that if you change the original, the clone is also modified accordingly. As an example, consider creating a flower with eight cloned petals rotated around the flower's center; tweaking the original petal causes all the cloned petals to similarly change. This feature actually comes straight from the SVG spec, so is a capability required for SVG compliance, but nobody knows of any other drawing apps that have a drawing operation quite like this.

¹ Inkscape Current Release Notes for Upcoming Version 0.39, <http://inkscape.sourceforge.net/cgi-bin/wiki.pl?ReleaseNotes>, as of May 23, 2004.

Another feature still in progress is ECMAScript (better known as Javascript), currently being developed on a branch from the main codebase by Ishmal. This promises to provide document-level scriptability, including simple programmatic animation.

There's also a new Open Clip Art Library² established to collect and promote SVG clip art for use in any of the open source drawing tools, and we hope to build in Inkscape strong support for browsing and using this library.

The latest contribution that I think will have widespread and exciting ramifications was brought to Inkscape quite out of the blue by Mike Hearn. Mike's project, called AutoPackage³, seeks to solve the perennial problem of easily installing software on Linux. It wraps the underlying RPM, Debian, etc. systems with a friendly GUI front end, similar to what's used on Windows. Mike's hoping Inkscape can help be a good proof of concept for his work, and we're looking forward to gaining an extremely easy installation mechanism for non-technical users.



I've worked on a lot of projects that have gotten few, if any, contributions from others, so I know just how remarkable and invaluable each of these contributions are. To me, this type of sharing is what makes Open Source so cool. Fred's contribution was the first of many such contributions for Inkscape. And who knows what surprise contribution will show up in our mail boxes tomorrow!

Thanks to all the 'scapers who lent an eye to reviewing this writing.

2 Please see: Open Clip Art Library, <http://www.openclipart.org>

3 Please see: AutoPackage, <http://www.autopackage.org>