RANDOM POINTS

LEWIS GRAHAM

Open Source Mania

In the last Random Points, I rambled on about something I am marginally qualified to address—optimal design. Continuing in this same vein let me discuss so-called "Open Source" software. The impetus for this is a sudden increase in confused noise regarding work done by the American Society for Photogrammetry and Remote Sensing (ASPRS)'s LAS Working Group.

The LAS Working Group (LWG) is a committee of the ASPRS LIDAR Division. The LWG is responsible for maintaining and updating the ubiquitous LAS point cloud format specification (LAS is not a standard—it is a file format). One member of the committee has recently been vocally advocating that his compression algorithm and resultant compression format be adopted as a "standard" by the industry. His actual motivation is that he has written a very large collection of LIDAR tools (which he sells) that consume and produce this compressed format. If the world adopts his compression format as a 'standard,' imagine the market for his tools! I think this is an exceedingly clever business strategy. The trigger for this sudden activity is that ESRI has released a free compressor/decompressor as well (albeit sans source code). If the ESRI compressor takes off, there goes the clever business strategy!

Of course, in my mind, all of this has nothing to do with the LWG; we just maintain the base LAS format. If someone wants to compress/decompress, that's fine. We (as the LWG) would just say "make sure that the file is identical after the round trip compress/decompress if you want it consumable by software that reads files that adhere to the LAS specification."

But all of this discussion did lead me to think a lot about "open source" software. I quote "open source" because it is one of those terms that is wielded like a weapon but is actually not defined. What exactly is "open source" software, anyway? I think it basically means that the source code is somehow available in a form that can be ingested by a software development system. So for the remainder of this article, I will refer to "open source" code as OSC with the quotes around "open source" inferred. OSC does not mean free (as in no money changing hands) in spite of the name of the anchor organization; the Free Software Foundation.

> "Using the wrong open source licensing can be equivalent to injecting your company with an intellectual property time bomb."

Now the reality is that you may not be able to do anything with that OSC for several reasons. The most onerous is licensing. Many, many folks post and consume OSC without really understanding the license that they attach (in the case of being a producer) or, most dangerously, they inherit by use of OSC. Given the caveat that getting legal advice from this column is a really bad idea, let me point out a very serious problem. The license that is the linchpin of the Free Software Foundation is the General Public License (GPL). This is a so-called "copy-left" license that requires the user of a GPL bit of code to GPL all "touching" code. GPLing code means making the source code available for free. Obviously any software subject to the GPL has an intellectual property value of zero. This does not mean the code cannot be sold; it simply means anyone else can sell the code as well. There are all sorts of arguments about what "touch" legally means. Because this is such a critically important issue, the GPL is the most complex and toxic license in the industry.

The Open Software Foundation devotes reams of (digital, thank God)

Displayed with permission • *LiDAR News Magazine* • Vol. 5 No. 4 • Copyright 2015 Spatial Media • www.lidarnews.com

pages to explaining what you can and cannot do with GPL software. At the end of the day, if you touch a piece of GPL code with the nine foot pole of launching it with a Python script, that script must now be GPLed. If anyone tells you this is not true, make sure that someone is a well-seasoned IP attorney who is very familiar with the enforcement actions taken by the Open Software Foundation in Europe (a climate much more sympathetic to socialized technology). No wonder Steve Ballmer referred to the GPL as a virus!

Here is the more dangerous aspect of GPL. GPL explicitly says that if your code touches GPL code in any manner at all (including the aforementioned CMD launch, a UNIX fork, especially any kind of dll and so on) all code is now GPL. However, the GPL says that if you do not make your program available outside of your organization, you do not have to make the source code available. A lot of folks read this to mean that, as long as I don't let anyone outside of my organization have any part of my software, my software is not subject to GPL. This is absolutely not correct and forms the core of the danger. Your code is 100% GPL—you just do not, at this time, have to release the source. The operative phrase here is "at this time."

Here is the rub. Let's say you are a commercial LIDAR company who has developed an entire set of production software that lets you process LIDAR data in new and much more competitive ways. Let's make it concrete. Say you use QGIS (www.qgis.org) as the core for general utilities such as display and so forth. You have no intention of providing your software to anyone outside your organization. You are in state of: all of your developed code (that stuff you spent $2,356,765 developing) is now legally GPL licensed (whether you say so or not). "No problem", you say, "we have no intention of trying to sell this software."

However, consider the following two scenarios. The first is you take on a subcontractor who is a company independent of your own. You decide to let them use your production software. You have just exposed your GPL code and now are obligated to release all of your source code. Would you get caught doing this and really have to release the code? Probably not.

But here is the killer (meaning bad, not good!) scenario—it is now time to change the structure of your company by selling to another company, merging, going public or some other transaction. Your GPL source code release obligation triggers due to this event. Again, will you get caught? That is not the point. If the acquirer (let's just use the scenario that you are being acquired) has a good IP person on their due diligence team (and, of course, they usually will have), this person will immediately see the problem and discount the value of your IP to zero. If the deal was predicated on the value of your IP, your deal just went south. Is this a real danger? You better believe it is. I wonder if Asia Air Survey, a gold sponsor of QGIS, is aware of this ticking time bomb?

"I am a university researcher and I think all software should be GPL!" Well, I won't go into the arguments about how somebody has to make some money to pay those salaries but rather point out one more grain of anthrax in the GPL. Suppose you have developed some very, very clever algorithm on which you and your university have applied for a patent. If you have coded your algorithm and used any GPL whatsoever, you just GPLed your patent. The patent rights effectively transfer to the Open Software Foundation for free distribution.

Well, I think you get the idea. The GPL is beyond toxic. What about the Lesser GPL (LGPL)? Code written against LGPL can also hit source code release triggers. LGPL is not as clear about the circumstances as is the GPL. The advice I get from our IP attorney is to avoid LGPL except in carefully crafted circumstances.

Does this mean the entire idea of Open Source Code is bad? No, not at all. My own company uses (and contributes back to) a lot of open source. For example, we joined the DotSpatial development group some years ago and used this foundation for our core client code in the GeoCue software suite. We contributed back to the DotSpatial source so code we wrote is off running in someone else's DotSpatial-based application. We did this work under the very reasonable MIT license. Sadly, DotSpatial moved to the LGPL licensing scheme in 2011. Of course, one of the best examples of extremely useful and hence used OSC is Geospatial Data Abstraction Library (GDAL). GDAL is used, among many other things, for reading and writing rasters and vectors. GDAL is licensed under the MIT scheme, making it useful for all applications.

The bottom line here is that OSC can be incredibly useful or incredibly toxic, depending on the license. Do not touch GPL without very good IP counsel (regardless of your purpose) and be very careful with LGPL. Perhaps if something drives me to another OSC rant, we can look at the economic side of OSC. ℹ

Lewis Graham is the President and CTO of GeoCue Corporation. GeoCue is North America's largest supplier of LIDAR production and workflow tools and consulting services for airborne and mobile laser scanning.