

Software Licensing in the Next Decade

ITAK Columnist

John Tomeny

IAITAM Fellow, Convener of the ISO/IEC 19770-3 SAM Standards working group of Software Entitlements, and VP of Business Development Sassafras Software Inc.



The rapid proliferation of mobile, virtual, and centrally managed computers drives the demand to invent new licensing models. Software licensing must be set free from the restriction of requiring an entitlement for an executable file simply because the file exists. There are numerous reasons for this; some of which we will explore in this article.

Nearly everything about a software asset management (SAM) practitioner's work today circles around the question of file existence. We conduct software audits to identify what is installed. Then we attempt to reconcile the results of our audits with our licensing status. We then uninstall files or 'true-up' purchases to correct our actual licensing position. And then next week, next month, or next quarter, we do this all over again. We have tools which can measure usage, or even enforce usage limits, but we cannot take full advantage of these tools when dealing with certain ubiquitous types of software licenses.

Our IT departments are chained to a never ending cycle of audit-reconcile-adjust licensing because software licensing language is fundamentally flawed from a SAM practitioner's point of view.

The most common desktop software licensing models in use today emerged in the 1980's, at a time when personal computing was pre-enterprise, pre-mobile, pre-virtual, and pre-"managed". These licensing models did not anticipate the evolution of IT "managed computing" practices for protecting

software assets and license compliance. As such, they are not well suited to fulfilling the typical requirements of today's SAM practitioners.

The most popular of these old models is per-device (or "node-locked") licensing. Per device licensing made sense in the 1980's. Early personal computers could only run software that was installed locally. There were no network deployment technologies (there were no networks). In order to install software, you had to go to a computer and install it. So you wouldn't bother unless you were going to use it on that computer. Thus per device licensing was tied to file existence.

The license model equates "installed" with "use". It served customers and software publishers well in the 80's because personal computers were isolated from one another. But it's been a bad fit since the earliest rise in computer networks in the 1990's. Personal computers in business are not isolated, but each one is part of a larger managed computing environment.

At the root of this problem is a fundamentally misdirected sense of value. Software license agreements are empowered by copyright law, which was originally written to protect the intellectual property of literary works. And so per-device software licensing emerged from the perspective that applied copyright law to software licensing in a manner "just like a book".

The popular belief of the 80's was that just like a book could not exist in two different places at the same time, neither

should a software program be used by two different people in two different places at the same time. The intellectual property that was (and is) being protected was the software executable code.

But the "just like a book" metaphor was ambiguous almost from the very beginning. As soon as companies adopted computer networks, the "book" analogy for per-device licensing lost its meaning. Once software could be accessed across a network, the software ("the book") could be used by different computers sharing access to a single executable file. File servers provided unlimited access to a single executable file and the technical means to enable simultaneous execution from multiple users.

While most publishers worked to shore up the definition of per-device licensing, some sought clearer meaning in "value-based" licensing models. Certain publishers recognized that they were trying to license and protect the wrong thing - the mere existence of software executable code; when instead, they should be licensing (and protecting) the usage rights to the executable code.

One such licensing model that was popularized in the 1990's did exactly that. Concurrent Use Licensing (CUL) focused on the value of using software rather than merely having software. In a sense, CUL grew from recognition that, unlike a physical book that could be protected through copyright law by controlling the number of copies, software (electronic in its nature) needed a different type of protection. To effectively protect its intellectual property, the use of software needed to be licensed, and not merely its existence.

Concurrent use licensing offered several advantages. One of those was the convenience to the customer of not having to worry about where the software was installed as long as there was a process in place to prevent overuse. Software licensing became an important product feature that influenced the value of a software product. But often publishers and customers alike were slow to understand

the relationship between value and cost - more on that below.

Since its introduction, CUL has emerged as a mainstream license model that continues to be offered today by hundreds of software publishers. Unlike the “audit-centric” approach of per-device licensing (which requires us to count files), CUL is an “Entitlement-centric” model that licenses the use of software.

But there are also shortcomings with the CUL model. First, it cannot effectively serve the growing mobile computer market because computers must have a steady communication link to a network-based license server in order to maintain compliance with the licensed limit. Secondly, CUL is sometimes a poor measure of value. A single license for a program that is used infrequently for very short periods of time can potentially serve thousands of users, while a license for a program used frequently for long periods of time would serve a much smaller number of users. Thus, value is hard to assign.

Still, the entitlement-centric approach of licensing the use of software offers a lot of promise for us going forward into this next decade. As managed computing environments continue to expand the use of mobile devices and virtual computing, we can already see how impossible it is to effectively enforce software license compliance if we are limited to using an out-dated, audit-centric licensing model.

Today, computing environments are rapidly evolving and are becoming more centrally managed. Enterprise networks with the capability to run software programs installed remotely on computers on the other side of the world are commonplace. Image management and deployment technologies can quickly install or uninstall software on thousands of computers. Virtual computers and virtual application technologies are rapidly expanding into mainstream use.

Virtual computers are nothing more than computer files themselves that can be created and destroyed rapidly - even before they can be audited. Or if they



are audited, the computers (and all the installed software) may be destroyed as soon as they are no longer in active use - and likely before the IT staff reads the report that includes them. So even when we are able to count file existence, the number we arrive at might be meaningless due to virtual computers that have been destroyed as well as back-up copies or temporarily installed demo copies that don't require licensing.

It has become irrelevant whether software is installed on the computer in front of you, or on a virtual server on another continent. Audit-centric software licensing, based on file existence, cannot keep up with these rapidly moving targets. Publishers' continued reliance on this antiquated license model may drive licensing costs higher, hinder the development of “optimized SAM” (the discipline of analyzing usage metrics to reduce software costs), and even inhibit publishers' ability to protect their intellectual property.

New entitlement metrics are needed to effectively support the growth of managed computing. Existing licensing policies frequently fail to provide a practical means for achieving compliance in corporate environments. These new metrics must be set free from file existence.

Further, software publishers need to embrace new licensing models before technology completely runs away from them, as it did for the music industry when MP3 technology arrived. Digging their heels in on audit-centric (file existence based) licensing will continue to make things worse for everyone. But the adoption of entitlement-centric licensing will be far more effective; both at protecting intellectual property and in serving managed computing. Paradigm shifts can be messy, but ultimately beneficial.

The obvious question, of course, is “How do we set software licensing free from file existence?” Since the primary goal of software licensing, from a publisher's point of view, is intellectual property protection, then it seems reasonable for publishers to take an interest in how many copies are created and distributed for each software program.

This issue brings to mind the story of my visit to the offices of the Software & Information Industry Association (SIIA) in 1999, which provides further clarity on the differences between audit-centric and entitlement-centric software licensing. 1999 was the year that the Software Publishers Association (SPA) merged with the Information Industry Association (IIA) to form the SIIA. The

purpose of my visit was to demonstrate a new technology that Sassafras Software had introduced for software auditing and discovery. We were all in a meeting room with representatives from both the “IIA” contingent and the “SPA” contingent of the newly formed SIIA while I demonstrated a software audit.

As we viewed the audit results, one of the SIIA staffers from IIA noticed that there were two copies of a certain program installed on one of the test computers. He immediately took issue with the “copyright violation” that was evident from more than one copy of the software on that single computer. And then the fun began as the SPA side of the house argued that it was not a software license violation, thus it was not a copyright violation. And, of course, the IIA contingent argued that it was indeed both a violation of copyright and of license.

At the end of the day, no one in the room agreed. And many of the arguments that flew through the room that day revealed just how difficult it is to shoehorn copyright law (originally written to protect literary works) into a computer. But years later, Adobe Corporation took some steps that helped to simplify the problem. At least for their own customers they were able to set aside the issue of file existence.

In 2006, Adobe introduced their ill-fated “Adobe License Manager” (ALM). Despite ALM suffering from poor market acceptance due to some implementation problems; use of the technology ushered in a software licensing model that I have described in the past as “activation-based licensing”. It was not an entirely new concept, but it was likely the first time that this model had seen such broad distribution.

Quoting from my last article on this subject: “Under ALM, [Adobe’s] activation-based license was granted to an installed copy of software when the software was given its serial number - or activation code. A central license server endowed the software with the ability to launch.” If at some time in the future, the customer wished to ‘harvest’ the license to redeploy it elsewhere, they

could use a process to remove the license activation, which would then render the software either incapable of launching, or capable of launching in demo mode only.

The uniqueness of this approach is that Adobe did not require the customer to uninstall the software when it was not licensed. It is important to see that the 1980’s “audit-centric” question of file existence was entirely set aside by Adobe’s model. File existence didn’t matter because a central licensing authority could be used to either assign or remove execution privileges or even access to specific product features. One benefit was that IT configuration managers could clone disk images containing the Adobe software to any number of computers and then select a subset of computers that they wished to grant licensing rights to for Adobe’s software products.

Looking back over the last two decades, software company product managers in the 90’s often seemed entirely unaware of software licensing issues. They were primarily focused on the development of new product features to meet the demands of their target markets. And the intimate relationship between licensing rights and product value was frequently overlooked as publishers offered high-value CUL at bulk-purchase discounted pricing.

In the next decade, the 2000’s, product managers became more enlightened in software licensing topics; viewing them as important contractual vehicles that drove revenue generation. But that view still remains too narrow; primarily serving the interests of the software publisher and not the customer. As we look toward the next decade, it will be important for product managers to view software licenses from a new perspective - as product features. Each set of specific rights granted by a software license has a certain value in the market place and therefore a specific competitive price can be applied.

Before we take this further to examine one possible solution, let’s review the problems we’re attempting to solve. The disadvantages of Per-device

licensing are that it cannot adjust quickly to reflect current installations in a rapidly changing environment, and that it is also a poor reflection of value - if we measure value by use. Unused software has no value to a customer, and so it follows that a license metric that measures file existence rather than use does not truly reflect value for the customer.

On the entitlement-centric side, the disadvantages of Concurrent Use Licensing are that it cannot manage entitlements for off-network computers, and is sometimes a poor measure of value.

And so, on one side we have the extreme of per-device licensing which is utterly impractical for customers. On the other side we have the extreme of CUL that risks undervaluing software when used as the metric for pricing products that are used infrequently.

What we need is something in-between these two extremes - a license entitlement supported by a central license server that builds on the advantages of the two extremes while diminishing their disadvantages. The entitlement should be granted to a computer on launch of a program just as with CUL, or on “activation” of the software just as with per-device licensing. And then it stays with the computer for some “lease” duration - beyond the actual usage of the program, but not indefinitely.

If during this lease duration the end-user re-launches and uses the software, the lease duration continues to be extended beyond the last time of use. So this model adopts modified behaviors of both CUL and per-device licensing.

Such a lease metric, when used as a replacement for per-device licensing and configured with a 30 to 90-day lease period, can effectively create a permanent, per-device, license for computers that regularly use the software. But it offers the advantage of releasing an entitlement from a computer that ceases to use the software after the lease period has expired. Then instead of the license being lost when a computer ceases to exist (as with a



virtual computer) or is disposed of at end of life, the lease timer expires on schedule and the license is returned to a central licensing pool where it becomes available for another computer.

The same metric, when used as an alternative to CUL and configured with a short-term, multi-day lease period, can effectively serve mobile computers that come and go daily or over the weekends, while still allowing fast, efficient release of licenses that are no longer in use.

Some software publishers have already begun to recognize the validity of this lease concept in per-device licensing by offering licensing clauses that permit the transfer of an entitlement from one computer to another after some period - typically 90 days. And Microsoft has recently announced plans to modify its software licensing program to reduce the costs of licensing in virtualized environments. So this is a good time to bring a new subject to the negotiating table.

Software entitlements based on lease timers offer many benefits for both

publishers and customers in our emerging world of enterprise computing. The lease model can move the IT industry forward by releasing us from the limitations of audit-centric licensing and replacing it with an entitlement-centric model that links directly to value based on a combined measure of both license allocation (from the per-device model) and software usage (from the CUL model).

The alternative, in the absence of new licensing models that are linked to value, will be to accept licensing language that is completely out of step with the realities of modern computing technologies and practices. Much like the adjustments being forced onto the music industry in the face of a technological sea change, the software industry must adapt its pricing metrics and licensing rights going forward.

Don't be content to believe you are obtaining value based solely on price negotiations. This leaves you vulnerable to uncertain liabilities due to licensing terms and conditions that simply don't

match reality no matter how diligent your compliance efforts may be. Arm yourself with a real understanding of your licensing requirements based both on historical usage tracking and on an understanding of the practical details of your deployment and management infrastructure. Also learn about the availability of central license server technologies that open up new measures of utility and new enforcement options. Be bold at the negotiating table and propose new licensing models that better serve your ITAM goals.

As always, I enjoy feedback, discussion, and collegial debate on software licensing topics. I invite you to contact me at johnt@sassafra.com. Important: in order for your email to safely make the trip through spam filters, please include the acronym "ITAK" in the subject of your message.

MobileReach

Increase database ACCURACY and worker PRODUCTIVITY



Improve Your Asset Management Processes

- Review Purchase Orders, Tag Assets, and create Asset Records right at your Receiving Location(s)
- Perform Physical Inventory much quicker and with greater accuracy
- Capture daily asset changes when and where they occur
- Manage Inventory in an accurate and timely fashion

Take Advantage Of Tried & True Solutions

- Capture data quickly and accurately with mobile barcode and/or RFID scanners
- Operate in a wireless, cellular, or disconnected environment
- Experience a highly functional and customizable User Interface
- Protect data integrity and privacy with a highly secure solution