Being Human in the Digital Age

Better Scripts, Better Games

The Evolution of Virtualization

ACM Fellows

An Interview with C.A.R. Hoare
We need to encourage system thinking that embraces hardware and user environment as well as software.

The System

The problems surrounding the six issues listed here are in large measure the consequence of an overly narrow view of the system for which the software engineer is responsible. Although controlled by software, the system is usually a complex combination of software, hardware, and environment.

Platform independence is an ideal of many software systems. It means that the software should work under a choice of operating systems and computing hardware. To achieve this, all the platform-dependent functions are gathered into a platform interface module, then, poring the system to another platform entails only the building of that module for the new platform. Examples of this are the Basic Input-Output System (BIOS) component of operating systems and the Java Virtual Machine (JVM). When this can be achieved, the software engineer is justified in a software-centric view of the system.

But not all software systems are platform independent. A prominent example is the control system for advanced aircraft. The control system is implemented as a distributed system across many processors throughout the structure where they can be close to sensors and control surfaces. Another example is software in any large system that must constantly adapt in a rapidly changing environment. In these cases the characteristics of the hardware, the interconnections, and the environment can strongly influence the software design. The software engineer must either know the system well, or must interact well with some one who does. In such cases adding a system engineer to the team will be very important.

Engineering Team

No matter what process engineers use to achieve their system objectives, they must form and manage an engineering team. Much has been written on this topic. Software engineering curricula are getting better at teaching students how to form and work on effective teams, but many have a long way to go.

Every software team has four important roles to fill. These roles can be spread out among several people:

1. The software architect gathers the requirements and turns them into specifications, seeks an understanding of the entire system and its trade-offs, and develops an architecture plan for the system and its user interfaces.

2. The software engineer creates a system that best meets the architecture plan. The engineer identifies and addresses conflicts and constraints, is misused by the architect, and designs controls and feedbacks to address them. The engineer also designs and oversees tests.

3. The programmer must have the knowledge and experience to design an economical and effective solution with a predictable outcome.

The programmer converts the engineering designs into working code. Programmers are probably owners in their own right because they must develop efficient, dependable programs for the design. Moreover, anyone who has been a programmer knows how easy it is to make mistakes and how much time and effort are needed to detect and remove mistakes from code. When the software engineer has provided a good specification, with known exceptions predefined and controls clearly delineated, the programmer can work within a model that makes the job of implementation less error-prone.

The project manager is responsible for coordinating all the parts of the system according to the schedule, getting the resources, and staying within budgets. The project manager interfaces with the project stakeholders, architects, engineers, and programmers to ensure the project produces value for the stakeholders.

In some cases, as noted previously, a systems engineer will also be needed on the team.

Conclusion

We have not arrived at that point in software engineering practice where we can satisfy all the engineering criteria described in this column. We still need more effective tools, better software engineering education, and wider adoption of the most effective practices. Even more, we need to encourage system thinking that embraces hardware and user environment as well as software.

By understanding the fundamental ideas that link all engineering disciplines, we can recognize how these ideas can contribute to better software development, and we will help us construct the engineering reference discipline that Glass tells us is missing from our software engineering education. Let us put this controversy to rest.

References


When you purchase a software package, the package will often inform you (or the software will inform you when you install it on your computer) that you are not the "owner" of a copy of it, but only a "licensor" whose entitlement to use the software is subject to certain restrictions. This may include a restriction on transferring your copy of that software to anyone else. Suppose you ignore the no-transfer restriction and sell the software to someone. Have you breached an enforceable contractual obligation to the software's developer? By transferring the package to someone else, have you infringed copyright or induced the purchaser of the software to infringe copyright? Is the purchaser of the used software an infringer when he loads it on his computer? Is he too bound by the license restrictions?

There has been, through, no definitive court ruling on these questions. In Vernor v. Autodesk, Inc., a judge recently ruled that a purchaser of used software could lawfully sell the package merely because he was entitled to the benefits of the "first sale" rule of copyright law. This rule provides that although copyright owners may control distributions of their work to the public, the first sale of a particular, odd to the public exhausts their control over any further distribution of that copy. The rule applies to all transfers of ownership, including gifts or bequests, but not to licenses.

Reinforcing the Vernor ruling was UMG vs. Audigo, in which a judge recently refused to enforce a restrictive legend forbidding recipients of promotional CDs from selling or otherwise transferring the CDs to other people. As in Vernor, the court ruled that it was lawful for Augusto to sell the used CDs on eBay under the first sale doctrine. UMG has already appealed the Augusto ruling to the Ninth Circuit Court of Appeals, which reviews lower court decisions from California and Washington where Vernor and Augusto were rendered, and Autodesk is likely to appeal as well. I predict that the Augusto ruling will be affirmed. Vernor is a closer case, but it may also be affirmed unless the Ninth Circuit overturns one of its long-standing precedents.

Legally Speaking

When is a "License" Really a Sale?

Can you resell software even if the package says you can't? What are the implications for copyright law in the Quanta decision discussed in the November 2000 column?
The software industry will likely weigh in heavily on the Augusto and Vernor cases, for the decisions challenge a long-standing industry practice.

One important incident of ownership is a right to an unlimited duration of possession, whereas an incident of a license is an expectation that the property will be returned to its owner when the license expires or is breached. The recipts of the CDs seemed to be entitled to keep the CDs, and UMG probed no evidence that it expected to resell the CDs. UMG could do nothing, moreover, if recipients destroyed these CDs, even though this would exhaust UMG’s claimed property rights. Nor were the recipients of the CDs under any obligation to UMG to return the music.

The judge concluded that UMG’s shipment of the CDs was a gift to the recipients, not a license. The judge’s rationale was that the Supreme Court refused to enforce in Robbins v. The County of Los Angeles, which established the first license rule in copyright law.

But the sale of the CDs cannot reasonably be understood to have granted UMG’s restrictive legend. Indeed, they illustrated their lack of assent to its sale by selling or giving the CDs away. Because they were free to transfer the CDs to anyone, so was Augusto.

Consumers Union is submitting an amicus curiae (friend of the court) brief in Augusto pointing out that if the courts allowed the restrictive legends to bind consumers, it would encourage manufacturers of all types of goods embodying some patented or copyrighted innovation to adopt restrictive legends. Such a rule would substantially undermine competition in the marketplace for used goods.

Some defer to the widespread practice in the software industry of licensing software rather than selling it. If consumers have agreed to be licensed and the license forbids transfer of the software, moreover, third-party purchasers such as Vernor are arguably incapable of being “owners” of the software. One can generally not obtain broader property rights in an artifact than had the person from whom you got it.

Yet, other judges have agreed with Vernor’s second case on the grounds that a one-time payment of money for a package of mass-marketed software that gives the purchaser rights to use software for an unlimited duration should be treated as a sale, even if it may be subject to some restrictions. Unless the Ninth Circuit overturns the decision, Vernor may win the right to resell used software on eBay.

It is a separate question whether CTA breached a contractual obligation to Autodesk by transferring the software to Vernor. But even so, should Vernor be bound by the contract’s restrictions on transfers? It would seem not since he has not installed the software on his computer and has not agreed to its terms. A fundamental difference between contract rights and intellectual property rights is that the former bind only the parties to the agreement, whereas the latter bind the world. Besides, Autodesk chose to make the license nontransferable, so how could it bind Vernor or his customers? The Ninth Circuit may view Vernor as an ordinary guy trying to make a buck in the used goods market, rather than an infringer of copyrights.

The software industry will likely weigh in heavily on the Augusto and Vernor cases, for the decisions challenge a long-standing industry practice. (Negotiated licenses will be unaffected if the Ninth Circuit affirms both rulings.) It remains to be seen whether the Ninth Circuit will recognize as legitimate the interests of people like Augusto and Vernor and their customers in the existence of a market for used goods protected by copyright law.
The software industry likely weigh in heavily on the Augusto and Vernor cases, for the decisions challenge a long-standing industry practice.

One important incident of ownership is a right to an unlimited duration of possession, whereas an incident of a license is an expectation that the software will be returned to its owner when the license expires or is terminated. The court decided that the CDs were to be returned to the CDs, and UMG did not explain no evidence that it expected to repossess the CDs. UMG could do nothing, and if it had obtained the right to possess it for an infinite period and without an obligation to return it, even though it had not possessed UMG's claims against the recipients. Nor were insiders recipients of the CDs under any obligation to UMG to return the music.

What will the Ninth Circuit Court of Appeals do? The Ninth Circuit Court of Appeals has an easier first case because the restrictive license applied to the CD in some of the cases. The court decided that the Ninth Circuit was not in its jurisdiction to decide the first case because the software had been licensed, not sold, to CTA.

Although eBay initially succeeded in its appeal, Vernor told eBay that it would not sell the software lawful under the first sale rule. Autodesk contacted eBay and eBay to assert that the sell would infringe its copyright because the software had been licensed, not sold, to CTA.

Although eBay initially succeeded in its appeal, Vernor told eBay that it would not sell the software lawful under the first sale rule. Autodesk contacted eBay and eBay to assert that the sell would infringe its copyright because the software had been licensed, not sold, to CTA.

One can generally not obtain broader property rights in an artifact than had the person from whom you got it.

plained in my November 2008 column, the Supreme Court ruled that a patent owner's effort to restrict commerce in licensed technologies was inconsistent with patent law's first sale rule. The Court left open the question about whether purchasers of the technologies could be held liable for breach of contractual restrictions, but made clear that they were not patent infringers.

Vernor is a tougher first sale case than Augusto for at least two reasons. First, CTA had agreed to abide by some of the conditions of the AutoCAD license. More generally, there is a stronger basis for inferring an assignment to “licensee” restrictions when a purchaser of a software package clicks “I agree” to terms of a license when installing the software (although it is not clear from Vernor whether CTA had authorized the Autodesk programs). Second, the case law on whether the first sale rule applies to mass-marketed software is mixed.

Some judges have been persuaded that software developers should be free to contract as they wish with their customers who may return the software if they find license terms unacceptable. Some defer to the widespread practice in the software industry of licensing software rather than selling it. If customers have agreed to be licensed and the license forbids transfer of the software, moreover, third-party purchasers such as Vernor are arguably incapable of being “owners” of that software. One can generally not obtain broader property rights in an artifact than had the person from whom you got it.

Yet, other judges have agreed with the Vernor decision that a one-time payment of money for a package of mass-marketed software that gives the purchaser rights to use software for an unlimited duration should be treated as a sale, even if it may be subject to some restrictions. Unless the Ninth Circuit applies the Vernor decision, Vernor may win the right to resell used software on eBay.

It is a separate question whether CTA breached a contractual obligation to Autodesk by transferring the software to Vernor. But even so, whether Vernor would be bound by the contract's restriction on transfers. It would seem not since he has not installed the software on his computer and has not agreed to its terms. A fundamental difference between contract rights and intellectual property rights is that the former bind only the parties to the agreement, whereas the latter bind the world. Besides, Autodesk chose to make the license nontransferable, so how could it bind Vernor or his customers? The Ninth Circuit may view Vernor as an ordinary guy trying to make a buck in the used goods market, rather than an infringer of copyrights.

The software industry will likely weigh in heavily on the Augusto and Vernor cases, for the decisions challenge a long-standing industry practice. (Negotiated licenses will be unaffected if the Ninth Circuit affirms both rulings.) It remains to be seen whether the Ninth Circuit will recognize as legitimate the interests of people like Augusto and Vernor and their customers in the existence of a market for used goods protected by copyright law.