The Proper Scope of Copyright in Software and Tests for Infringement
The Proper Scope of Copyright in Software and Tests for Infringement

Michael Headley, Fish & Richardson P.C.  
headley@fr.com

Ronald Johnston, Arnold & Porter LLP  
Ronald.Johnston@aporter.com

Randall Davis, MIT Computer Science and Artificial Intelligence Laboratory  
davis@csail.mit.edu

Lothar Determann, Baker & McKenzie LLP  
Lothar.Determann@bakermckenzie.com

Felix Wu, Cardozo School of Law  
felix.wu@yu.edu
The Proper Scope of Copyright in Software and Tests for Infringement

Agenda

• Software Copyright: Roots, Scope, and Impact on Developers

• Technical Considerations

• The Cloud

• The Intersection of Copyright and Patent Law
Just When We Thought “Things Never Change” -- Will Lightening Strike Twice?

Ronald Johnston
Arnold & Porter
At the turn of the century...

- “Court decisions are, *generously stated*, in a state of *creative ferment* concerning the methods by which nonliteral elements of computer programs may be identified and analyzed for copyrightability.” (5th Cir. 1994)

- “Software developers have no adequate guidelines regarding what level of independent development is required to avoid copyright infringement.” Nimmer, today, relying on last century cases.
Proper scope of copyright protection

- Open issues
- Alternative approaches
- What we may know
- Practical observations
Open question:
At what level of abstraction is software protectable?

- Functional composite
- Detailed tasks
- Organization of modules, internal program interfaces
- Program behavior or sequencing
- Compilations
Open question:
External structure of program?

- Language, selection of commands, syntax, mnemonics
- Access protocols
- Sequencing (e.g., screen sequence)
- Behavior (user functions, inputs, outputs)
Open question: The test for determining what is protectable?

- Abstraction-filtration-comparison
  - Test or logical approach
  - The hard questions
  - What do you filter out
  - External constraints: author?

- Merger
  - Alternative ways to express
  - Categorical (e.g., *Lotus*’ “method of operation”)
  - Balance expressive and functional aspects
Open question:
How to consider compatibility/interoperability – if at all?

- External constraint, fair use, merger, etc.
- Who authored? Role of license?
- What kind of compatibility
  - Clone
  - Add-on products
  - User familiarity
Open question: Comparison of programs

- Substantial similarity v. virtual identity v. material element
- How to treat compilation following filtering of individual elements
Open question: Scope of fair use

- Reverse engineering/decompilation
  - Game cases
  - Software cases
- Role of interoperability/compatibility
- Role of licensing for extended use
What we probably know

- Analogy to other literary works for protection of non-literal elements
  - Other proposals

- Drawing idea/expression boundary ad hoc
  - Boundary may move where feels more utilitarian?
  - Baker v. Selden offers “scant guidance”

- Alternative ways to do “it” evidences expression

- Not protected if “dictated” by efficiency to accomplish “idea”
Repeated themes

- Detailed implementation
- Creativity v. standard
- Number of alternatives
- Choice not to copy
- Competitive significance
- Deserving v. non-deserving
A View from the Technical Side

Randall Davis

MIT Computer Science and Artificial Intelligence Laboratory
Comments

- Software is caught in a pair of procrustean beds
- Cross-cultural conversations create confusion
- Abstractions are pervasive and of the essence in software
- AFC works; SSO doesn’t
- The cloud is (mostly) irrelevant
Common knowledge

- Software is caught in a pair of procrustean beds
Common knowledge

- Software is caught in a pair of procrustean beds

  - *Software is a machine whose medium of construction happens to be text.*
Cross-cultural conversations create confusion

- 102(b)
  - …procedure, process, system, method of operation

- Every one of these is a well-understood term of art in computer science.
  - I know what they mean in my lab
  - I don’t know what they mean in the courtroom
  - Does anyone?

- When are two algorithms equivalent?
  - Underspecified question
  - For what purpose, on what dimension
Abstractions are pervasive and of the essence

- That’s how non-trivial programs are created
- Abstractions come in layers
- It’s abstractions all the way down…
Abstractions II

- Good abstractions are *valuable, carefully selected, compilations*

- Each abstraction layer is a *language*
  - Build on the layer below, built on by layer above
  - Languages should be available for anyone to use
Abstractions III

- Each abstraction layer is an *interface* (for the code beneath it)

- Some interfaces are *intended* to be used by others
  - Languages (Java), operating systems

- Other interfaces are used because they succeeded
  - Sega
  - Changes over time
Abstractions IV

- Each level of abstraction is an interface to the code below it

- What level of abstraction is protectable?
  - Is only the top one uncopyrightable? Oracle v Google
  - Is only the publically disclosed one uncopyrightable?
    - But: Sega
Abstractions V

- If
  - Levels of abstraction are pervasive in software, and
  - LOA = (AP)I, and
  - interfaces are MOPs, then
  - what (in principle) becomes of software copyright?
AFC works, SSO does not

- **Empirical comment:** AFC works

- **In principle comment:** SSO does not
  - In software, *structure and sequence are independent*
  - Textual layout of code does not determine sequence of operation
  - Textual layout can be arbitrarily changed without affecting sequence
  - SSO is technically incoherent
The cloud is (mostly) irrelevant

- What is “the cloud”?  
  - Someone else’s disk drive…  
  - … that you don’t own and have no control over, and  
  - … that has been set up to be accessible from anywhere on earth.

- A small in-principle point  
  - What if we all went to computation cafes to get computing done?

- The cloud is (just) the delivery vehicle

- It seems the core issue is that someone else owns the machine.
Software Copyrights in the Cloud
From sales to licenses to services – the journey to the cloud

The basic premise was ....
- vendor receives upfront license fees
- customer receives software copies, updates
- customer keeps work product, data

The basic premise becomes ....
- vendor receives recurring license fees
- vendor keeps software copies, updates
- vendor receives work product, data
The promise to customers:
- price, cash flow
- quick deployment
- dynamic computing resources
- mobility
- outsourcing of hardware, software, change management
- security
The benefits for vendors:
- control over code
- no ‘distribution’ for purposes of open source license terms
- no ‘sale’ for purposes of exhaustion
- reverse engineering becomes difficult
- protection against unwanted interoperability via computer interference laws (CFAA, etc.)
What is it commercially? The cloud ...

The commercial character of SaaS, PaaS, IaaS:
- services elements
- lease elements
- licenses
- sales
Software Copyrights in the Cloud

What rights are infringed / need to be licensed?
– reproduction
– adaptation?
– distribution?
– public performance?
– public display?
– use?
Contracting re: Software Copyrights in the Cloud

- Limited need for licenses
- “Cloud privacy”
- Compliance monitoring, enforcement
- Data privacy, security obligations – allocation of duties and liabilities
- Data usage by vendors, big data
- Liability for hosted content, data
- Term, post termination transition
The new landscape

- Interoperability under copyright law (including DMCA) and computer interference and trespass laws
- Reverse engineering
- Open source license terms
- First sale doctrine
Software Copyrights – First Sale Doctrine

The new landscape

- 17 USC 109 – owner of copy may resell
- 17 USC 117 – owner of copy may reproduce/modify as necessary to use / maintain software

- Is licensee = owner?
  - no, according to Vernor v. Autodesk
  - yes, according to Oracle v. Usedsoft
International Exhaustion under U.S. Copyright Law

*Kirtsaeng v. John Wiley & Sons, Inc.*, United States Supreme Court, March 19, 2013
Software Copyrights in the Cloud – Quo Vadis?
Infringing Algorithms

Two Worlds Intersect

Prof. Felix Wu

Cardozo School of Law
Copyright Meets Patent

- Copyright infringement
  - Central claiming & substantial similarity
  - Abstraction-filtration-comparison test

- Patent infringement
  - Usually peripheral claiming
  - But not means-plus-function claims
Means-plus-Function Claims

  “An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.”
Lessons for Patent Law

- Abstraction-filtration-comparison
  - For means-plus-function claims?
Lessons for Copyright Law

- Abstraction-filtration-comparison
  - For means-plus-function claims?
  - Instead of for nonliteral copyright infringement?
Questions?

Michael Headley, Fish & Richardson P.C.  
headley@fr.com

Ronald Johnston, Arnold & Porter LLP  
Ronald.Johnston@aporter.com

Randall Davis, MIT Computer Science and Artificial Intelligence Laboratory  
davis@csail.mit.edu

Lothar Determann, Baker & McKenzie LLP  
Lothar.Determann@bakermckenzie.com

Felix Wu, Cardozo School of Law  
felix.wu@yu.edu