



The Software Attic

The case for expansive collecting

Len Shustek

May 5, 2006

Why collect?¹

- Our civilization runs on software
- No comprehensive and intentional activity has yet been undertaken to preserve software artifacts
- The authors are still alive, the objects are still available
 - Time is our enemy!

¹From Grady Booch, at the *Workshop on the Preservation of Classic Software*, Computer History Museum, October 16-17 2003

Open wide the software gates

- Collect with minimum censorship because
 - We don't know what will be important
 - We have the space
 - A large collection enables evolutionary and statistical studies
 - A large collection ensures diversity and comprehensiveness of the collection
 - It can be accomplished with a collaborative community

- We don't know what will be needed
 - “The historical utility of original artifacts is that they are available for interrogation in the light of unforeseen enquiry” – Doron Swade



Napoleon's Coat of the Chasseurs
a Cheval

- Space is not an issue
 - At most 300 GigaLines of source code have been written by human beings
 - 2M coders x 5K lines/yr x 30 years
 - 10% (3M) of all the programs ever written could be stored in a terabyte (\$400 at Fry's)
 - Much of it is self-identifying
 - Much of it is junk

Collect the source!

- A cultural artifact
 - a form of literature (D. Gabriel)
 - beautiful programs are works of art (D. Knuth)
- A view into the mind of the designer
 - intentions, assumptions, abstractions, mistakes, humor
 - little of this gets captured in any written form
- This is the embryonic first 50 years of millennia of software development
 - The transition from cave painting to impressionism

– A voluminous source repository can be analyzed to teach us about the evolution of software engineering

- architectural evolution
- data structure design
- use of algorithms
- optimization (and premature optimization)
- locality of function
- information hiding
- coding style and idioms
- defensive programming styles
- software redundancy
- failures and bugs
- module decomposition
- joint authorship
- programming language use

– The parlor is an unrepresentative sample

But also...

- Collect the binaries!
 - For use on restored, reconstructed or simulated old computers
- Collect the documentation!
 - manuals, notes, papers, email
- Collect the stories
 - interviews, reminiscences, websites

Other uses for the attic

- an archeological midden for future additions to the Parlor
 - now or later
- a legal resource for discovering and documenting prior art for software patents
- an historical resource for establishing credit and understanding influences

Tough issues

- copyright and ownership
- protecting trade secrets
- provenance
- reading and interpreting the bits (“transcoding”)
- insuring completeness
 - libraries?
 - program development environments?
- simulation difficulties
- permanence

Proposal:

“Global Software Repository”

- fueled by community contributions
 - web-based submission
- filter to keep signal-to-noise ratio high
 - panel of judges
- accessible to the limits of the law
 - mechanism for escrow
- limited metadata, provided by submitters
 - Depend on Google-like searches
- goal: 10,000 programs in two years