An Emulator for my Old System Today and Tomorrow?

Thoughts and guidelines about digital preservation by/of emulators

PONSARD Christophe
FOSDEM 21 - February 6-7 (online)
Context – NAM-IP Computer Museum

• Located in Namur/Belgium - 30’ from Brussels (worth a visit next FOSDEM when we are back at ULB !)

• Missions:
  • Preservation: safeguarding digital heritage, focus on local pioneers
  • Acquisition of artefacts, enriching collections
  • Exhibition: for all, specific animation, permanent/temporary
  • Research: about machines, software, communities

• “Container design”, an historical parallel

www.nam-ip.be
Preservation Constraints

• Machine in preservation mode
  • Difficult to be powered up
e.g. inDATA DAI rare Belgian computer [1978] (= our use case)
  • How to keep machine expertise?
  • How to transfer programs?
  • How to show running software experience about that machine?

➔ benefit of using an emulator, e.g.

• Some questions triggered:
  • Where to look for emulators
    (MAME/MESS, specific development, javascript ports...)?
  • What are interesting usage scenarios?
  • How to select one for some usage context?
  • As emulators are themselves part of history: How to make sure/contribute to the sustainability of those nice piece of software on the long run?

➔ Move from physical to digital preservation
Disclaimer / Focus

Not claiming to be expert, still discovering that area

Sharing some thoughts about:
• Past: digital legacy – what do we want to preserve?
• Present: what are the current strategies, what we can use?
• Future: how to make sure we keep that heritage for next generations?

Focus on emulators for retro-computing

Not looking at legal issues

Certainly partial /biased view ➔ feedback welcome
Outlook

• A step backward: digital preservation techniques
  • What is interesting for retro-computing?

• A wider look at emulators
  • A little analysis based on a timeline

• Use case and preservation tool chain

• Long term preservation strategies for emulators themselves

• Open Discussion
Digital Preservation

“maintain digital objects accessible and usable in an authentic manner for a long term into the future”

• Digital object: large scope
  • texts documents, emails, images, movies, music, spreadsheets
  • *programs, games*
  • entire web domains, social media…

• Usable:
  • Reading document, running code
    • Physical and logical threats: media not corrupted, format still supported
  • As usable, more usable ?
    e.g. better sound/screen resolution now, access time... (no tape loading please ;-

• Authentic
  • Same content for sure: semantic preservation
  • Same experience as in the past ? ("retro" dimension)
## Some Digital Preservation Strategies

do they work for retro-computing?

<table>
<thead>
<tr>
<th>Method</th>
<th>Idea</th>
<th>Example</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standardisation</td>
<td>Standard are there for a long time</td>
<td>Running on any IBM PC compatible</td>
<td>Few standard in retro time How long is “a long time”</td>
</tr>
<tr>
<td>Total preservation</td>
<td>Keep everything in state</td>
<td>Recap computer, change drive belt, refresh floppy disks…</td>
<td>Costs/expertise for old HW, old media Restricted usage</td>
</tr>
<tr>
<td>Encapsulation</td>
<td>Container with useful meta-data, (links to) software</td>
<td>Format information to decode image Instructions how to run a program</td>
<td>Documentation, access Not solving anything</td>
</tr>
<tr>
<td>Extraction</td>
<td>Mining useful stuff</td>
<td>Text without images, decompilation</td>
<td>Partial, fallback in degraded mode</td>
</tr>
<tr>
<td>Migration/media</td>
<td>Transfer to more stable/accessible media</td>
<td>From tape/floppy to HD/SSD/Cloud (wav/binary files)</td>
<td>Lost media specificities (or use disk image) Lifespan of new media ?</td>
</tr>
<tr>
<td>Migration/backward compatibility (interoperability)</td>
<td>Can read/run previous version on current applications</td>
<td>Open doc/check/save Run win32 application on win64</td>
<td>Only document can be saved in new format Possible loss, progressive degradation Limited in time</td>
</tr>
<tr>
<td>Emulation</td>
<td>Keep digital resource in original (logical) form but emulate hardware</td>
<td>MAME for games MESS for old micro-computer DosBOX for DOS programs</td>
<td>Need to write/maintain emu. ➔ see later How to access data ➔ see migration How perfect is the emulation</td>
</tr>
<tr>
<td>(Universal) Virtual Machine</td>
<td>Ensure independence wrt host platform using simple to implement VM</td>
<td>Historical UVM concept Java ? ➔ Javascript target</td>
<td>UVC targeting document not programs (see later in presentation)</td>
</tr>
</tbody>
</table>
Emulation – Background / Reminders

• **Definition**: HW or SW that enables one computer system (called the host) to behave like another computer system (called the guest).
  
  *Kind of “Digital Twin” of the computer system*

• **Technical goal**: enables the host system to run *unchanged* software or use peripheral devices designed for the guest system
  
  ≠ virtualization : CPU is emulated *(different instruction set)*

• **Usage scenarios/goals**: have a look at a timeline to discover them!
  
  • “Past”
  • “Present”
  • “Future”
Emulation - Background / Reminders

• **Definition**: HW or SW that enables one computer system (called the host) to behave like another computer system (called the guest).

• **Technical goal**: enables the host system to run software or use peripheral devices designed for the guest system.

• **Usage scenarios goals**: see next timeline to discover them!
  - **Past**: digital preservation, retro-computing, retro-gaming – e.g. MAME
  - **Past ➔ present**: backward compatibility against tech. change – e.g. Apple migrations
  - **Present**: compatibility with market (de facto) standards – e.g. PC emulators
  - **Present**: cost reduction (with performance trade-off) – e.g. FPU emulator
  - **Future**: develop new system before HW is available – e.g. ALTAIR BASIC!
Retro-computing Use Cases

• Retro-gaming

• Using specific software
  • E.g. non wysiwig text processor used by some writers like Wordstar by Robert J. Sawyer (SF writer) and R.R. Martin (Game of Thrones)

• Accessing old files

• For computer history
  • Study of past computer systems, recovery
  • Interactive experience look and feel/UI for public (Video is passive)
## Emulator Selection Criteria

<table>
<thead>
<tr>
<th>Criteria</th>
<th>The bad</th>
<th>The good</th>
<th>Hints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of installation configuration</td>
<td>Need to compile, find ROM, configure keyboard, ...</td>
<td>Platform bundles/pre-configured</td>
<td>Multi-system more difficult Easy All-in-one distribution on RPI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Running in browser [e.g. Internet Archive]</td>
<td></td>
</tr>
<tr>
<td>Ease of use</td>
<td>Raw emulator, poor media management</td>
<td>Power utilities integrated (media mgt, snapshots, debugger...)</td>
<td>Machine/constructor specific can bundle utilities Providing libretro API</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nice front-end</td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>Many component abstraction</td>
<td>Circuit/Cycle exact (but higher CPU)</td>
<td>Check forum, game compatibility</td>
</tr>
<tr>
<td>Long term support</td>
<td>Recent project, closed or small community, exotic technology</td>
<td>Long history, large community VM approach</td>
<td>Check repository activities OpenHub statistics</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Can use multiple actually, using shared media
- But not too many... encapsulation is poor, need to keep track of specific info...
- one multisystem [unified management]
  + one more specific [if useful as complement]
Compare for Amstrad CPC – MAME vs JavaCPC

Generic emulator
Need to install ROMS, configure keyboards...
Accuracy?

Open Source Metrics (OpenHub) - In a Nutshell, MAME...
... has had 77,136 commits made by 498 contributors representing 10,529,234 lines of code
... is mostly written in C++ ➔ compile approach with an average number of source code comments
... has a well established, mature codebase maintained by a very large development team with stable Y-O-Y commits
... took an estimated 3,239 years of effort (COCOMO model) starting with its first commit in December, 2007 ending with its most recent commit about 1 month ago

More specific (based on JEMU but only limited targets)
Lot of features, everything preconfigured, copy/paste, printer...
Accuracy?

In a Nutshell, JavaCPC - Amstrad CPC emulator in JAVA...
... has had 30 commits made by 3 contributors representing 38,957 lines of code
... is mostly written in Java with a low number of source code comments ➔ VM approach
... has a codebase with a very short history maintained by nobody with stable Y-O-Y commits
... took an estimated 10 years of effort (COCOMO model) starting with its first commit in December, 2008 ending with its most recent commit over 11 years ago

DSK exchange
Where to Look for Emulators?

Some entry points...

- **Multi-systems/platform/companies**
  
  [https://emulation.gametechwiki.com/index.php/Multi-system_emulators](https://emulation.gametechwiki.com/index.php/Multi-system_emulators)
  
  Inc. MAME, OpenEMU, BizHawk

- **Specific emulators**
  
  
    
    [https://www.planetemu.net](https://www.planetemu.net) [http://www.emu-france.com](http://www.emu-france.com) [French]

- **Retro-gaming zero install on raspberry pie**
  
  - Retro-pie: [https://retropie.org.uk](https://retropie.org.uk)
  
  - Recalbox: [https://www.recalbox.com](https://www.recalbox.com)

- **Web-browser**
  
  - Internet Archive: [https://archive.org/details/historicalsoftware](https://archive.org/details/historicalsoftware) (and others)
  
  - PCjs machines: [https://www.pcjs.org](https://www.pcjs.org) (key applications, OS available)

  - TinyEmus: [https://floooh.github.io/tiny8bit](https://floooh.github.io/tiny8bit)

- **Fan pages**
  
Building a Preservation Tool Chain – DAI Case

(type) scan → Keyboard scripting → MAME

Type on DAI keyboard → WAV2DAI → DAIBOOT (RS232)

Sound recording → WAV2DAI → DAIWAV → DAIBOOT (RS232)

Use tape HW → DAIBOOT (RS232)

Image dump (omnidisk) → (tape interface) → SD-DOS drive

Use floppy HW
DEMO

PACMAN

recovered

DAInamic

version

DAI Personal Computer
1978 Data Applications International
Driver: dai.cpp

CPU:
Intel 8080 2 MHz

Sound:
3×Speaker
DAI Custom Sound
Cassette

Video:
1056 × 302 (H) 50 Hz
Long Term Preservation of Emulators

Emulators
• digital artefacts so also part of the problem
• depend on the guest system

Different approaches to manage this long term evolution
• Migration = recompiling to new guest
• Emulation chains = stacking on emulated guest
• (Universal) Virtual Machine = isolate from guest

Can be combined/mixed!
Migration Approach = recompiling emulators

IDEA: recompile to new target
• Need community for recompiling + possible adaptations
• Same process for evolution and improvement
• Preserving performance, reliability through testing
• Frameworks have more chance to evolve than specific emulator
Emulator approach ➔ stacking emulators

IDEA: older system becomes emulated too
• Not same team working of different emulators (e.g. hidden win32 layer)
• Performance impact ➔ compensated by HW improvement
• Reliability impact ➔ more likely to accumulate
• Can be hidden/multiplatform (e.g. QEMU)
• Old emulator maintained (?) independently

• Alternative: virtual machine
  • If same CPU
  • Better performance
(Universal) Virtual Computer Approach

• IDEA: use VM technology to isolate from guest OS/HW
• Example: Java (JEMU, CPCEMU)
• Need to make sure new VM is developed/available for version “X” (bycode evolution ➔ backward compatibility)

• More elaborated approach: Universal VM
• Simple VM easy to re-implement when necessary
• More designed for accessing document artefacts than for code. Not practical for emulators.
To Conclude

• Emulators are great tools for many purposes
  • Great tool for museums at technical level
  • Easy/Zero install retro-computing/gaming help preserving collective memory

• Hope you enjoyed the quick journey in digital preservation and emulators

• May seem funny but not so easy to capture the full picture
  • Open Question: maybe need some knowledge base (where/how/who...) ?

• Feedback / ideas / contributions welcome !
References

**Papers:**

• Stewart Granger, Emulation as a Digital Preservation Strategy, D-Lib Magazine, ISSN 1082-9873, Volume 6 Number 10, October 2000


• Andreas Rauber, An Introduction to Digital Preservation, IFS, Vienna, 2013

• David S.H. Rosenthal, Emulation & Virtualization as Preservation Strategies, a report commissioned by The Andrew W. Mellon Foundation, New York, October 2015

• Cochrane, Euan, Jonathan Tilbury, and Oleg Stobbe. "Adding emulation functionality to existing digital preservation infrastructure." Journal of Digital Media Management 6.3, 2018

**Web:**


• [https://kaluszka.com/vt/emulation/history.html](https://kaluszka.com/vt/emulation/history.html)

• [https://www.zophar.net/articles/art_14-2.html](https://www.zophar.net/articles/art_14-2.html)

• [https://blog.dshr.org/2016/07/talk-at-jissccni-workshop.html](https://blog.dshr.org/2016/07/talk-at-jissccni-workshop.html) (blog of D. Rosenthal)