FOSDEM

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

Computer Museum NAM-IP

www.nam-ip.be

- 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

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 th
    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 /biaised 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 ➔ often data dimension
  - **programs, games** ➔ **more our focus**
  - entire web domains, social media… ➔ data link dimension

- 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 ?

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

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"**

Emulator Timeline (main milestones)

1962
Emulation Theory [IBM]
Larry Moss "Father" of emulation

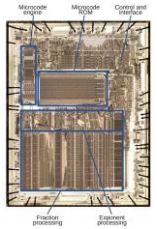
1965
IBM 360 emulator for "old" 7070

1975
ALTAIR BASIC 8008 emulated PDP-10

1982
80286 with 8086 real mode (FW emulator)
Baby blue CPMemulator

1985
Amiga SW Transformer to IBM-PC
1986
PCDitto for Atari ST

1989
FPU emulators



1990/91
First console emulators (e.g. NES)

1995
Virtual Game Boy
First OSS emulator

1996
EMU
First arcade Videogame emulator

1997
start of MAME

PSemu (early)

1998
start of MESS

2002
DOSBOX

2006
Mac Rosetta PPC➔Intel

2007
QEMU (OSS)
VirtualBox

2012
emscripten project

2015
MAME MESS merged

2021
Mac Rosetta II
Intel ➔ ARM

2013-2014
Internet Archive using JMESS/DOSBox in web browser

Emulation - Background / Reminders

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

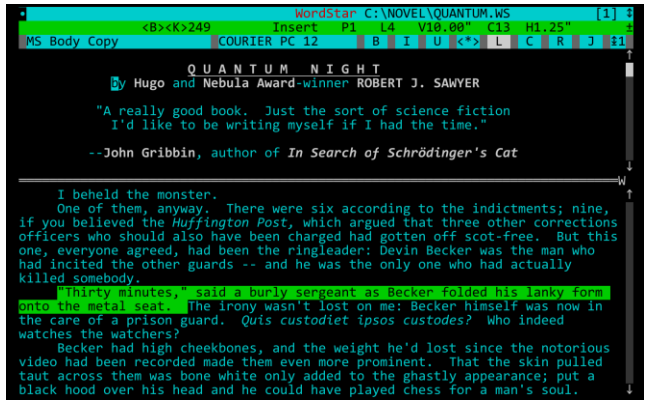
- **Technical goal**: enables the host system to run software or use peripheral devices designed for the guest system

- **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

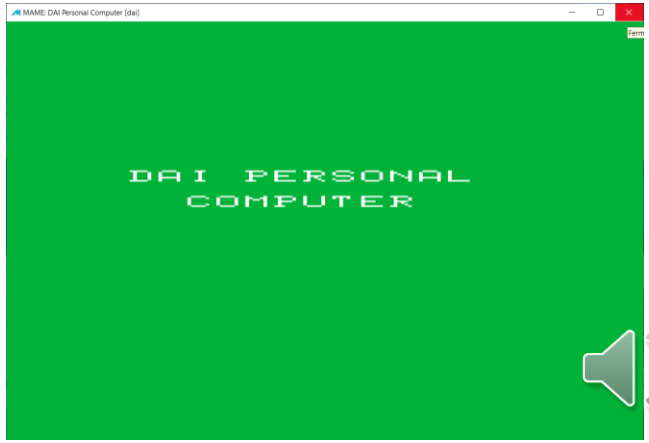
- Retro-gaming

- 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 ?

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

➔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 ?

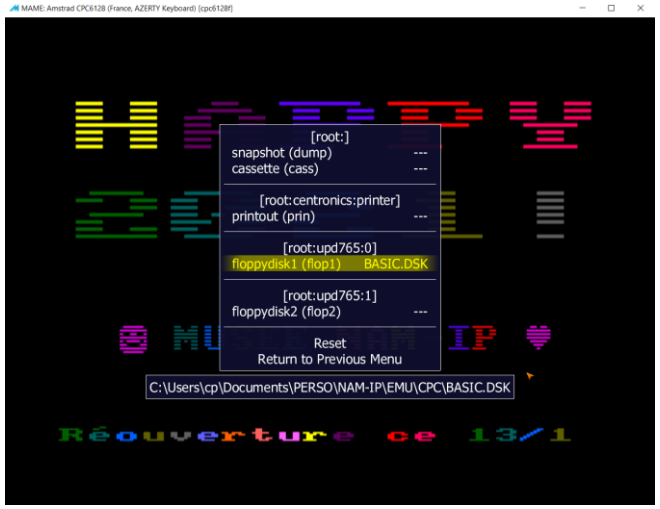
Open Source Metrics (OpenHub) - In a Nutshell, MAME...

... 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



DSK exchange

More specific (based on JEMU but only limited targets)

Lot of features, everything preconfigured, copy/paste, printer…

Accuracy ?

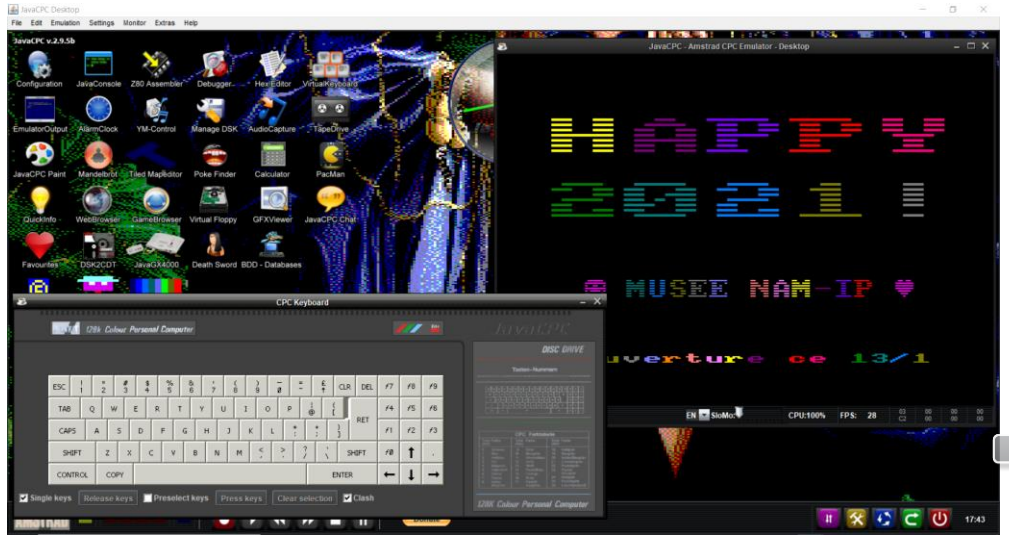
In a Nutshell, JavaCPC - Amstrad CPC emulator in JAVA...

... has had 30 commits made by 3 contributors
representing 38,957 lines of code

... is mostly written in Java ➔ VM approach
with a low number of source code comments

... has a codebase with a very short history
maintained by nobody
with stable Y-O-Y commits

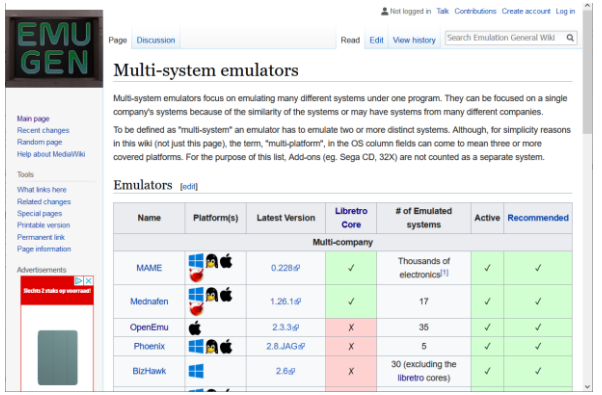
... took an estimated 10 years of effort (COCOMO model)
starting with its first commit in December, 2008
ending with its most recent commit over 11 years ago

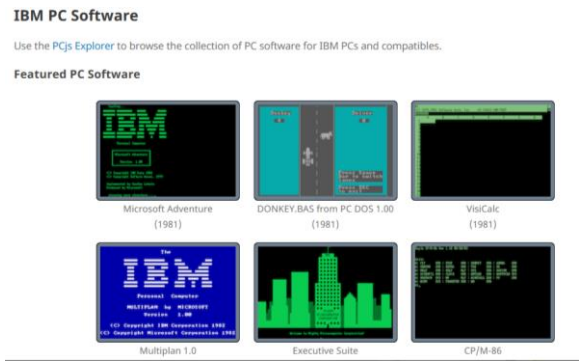


Where to Look for Emulators ?

Some entry points…

- Multi-systems/platform/companies
  https://emulation.gametechwiki.com/index.php/Multi-system_emulators
  Inc. MAME, OpenEMU, BizHawk
- Specific emulators
  - Computers: https://en.wikipedia.org/wiki/List_of_computer_system_emulators
  - Consoles: https://emulation.gametechwiki.com/index.php/Main_Page
    https://www.planetemu.net http://www.emu-france.com [French]
- Retro-gaming zero install on raspberry pie
  - Retro-pie: https://retropie.org.uk
  - Recalbox: https://www.recalbox.com
- Web-browser
  - Internet Archive: https://archive.org/details/historicalsoftware (and others)
  - PCjs machines: https://www.pcjs.org (key applications, OS available)
  - TinyEmus: https://floooh.github.io/tiny8bit
- Fan pages…
  - http://bruno.vivien.pagesperso-orange.fr/DAI/programmes/index.htm





ParaDAI

Building a Preservation Tool Chain – DAI Case

```
10      MODE 0:GOTO 800
20      F=SIN(X)*COS(Y)
30      IF F>Z2 THEN F=Z2
40      IF F<Z1 THEN F=Z1
50      RETURN
```

(type)
scan

BAS

Keyboard scripting

MAME

Type on DAI keyboard

Sound recording

WAV

WAV2DAI

DAIWAV

BIN

DAIBOOT
(RS232)

Use tape HW

image dump
(omnidisk)

IMG

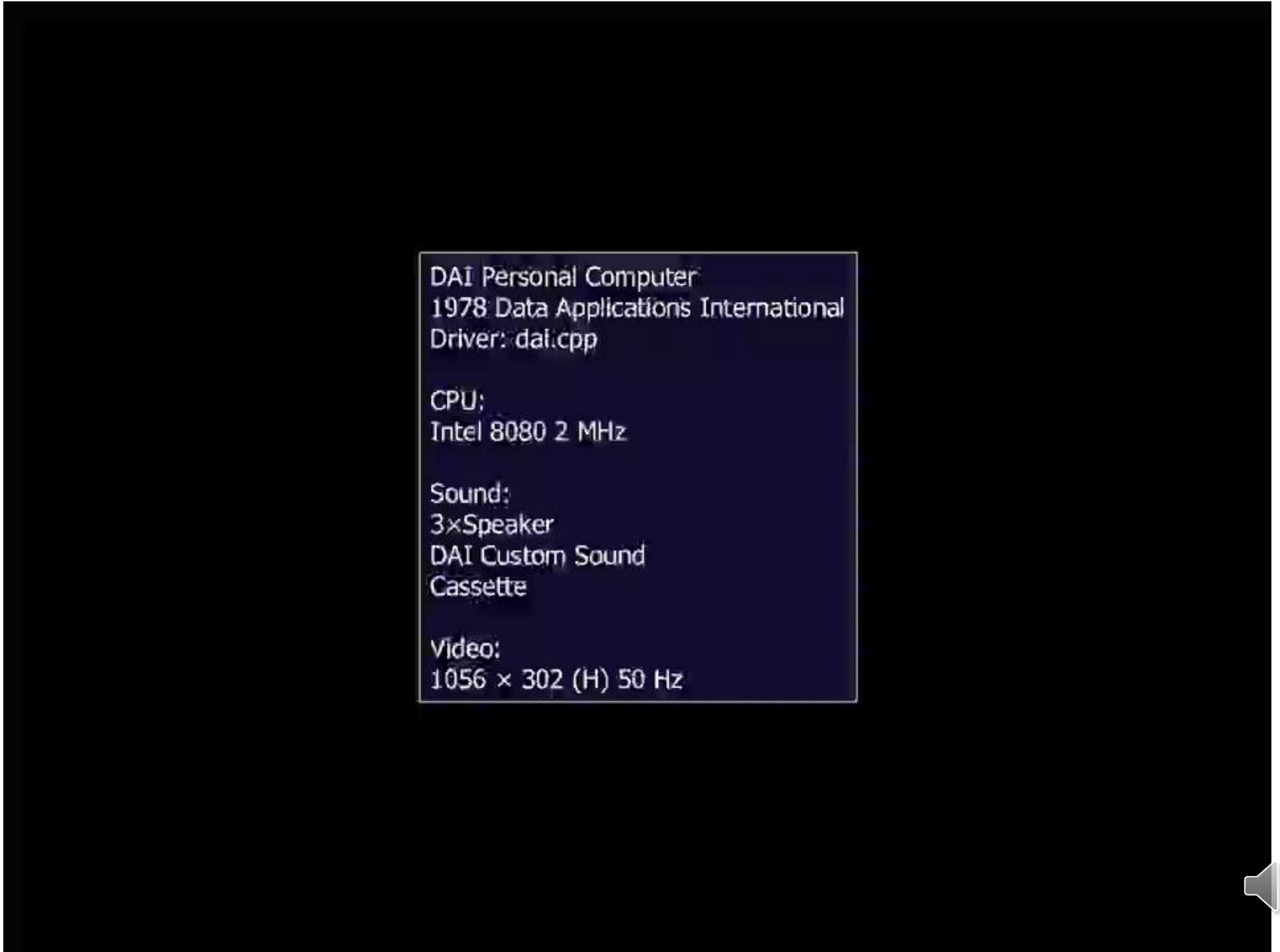
(tape interface)

SD-DOS drive

Use floppy HW

DEMO

PACMAN recovered DAInamic version



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

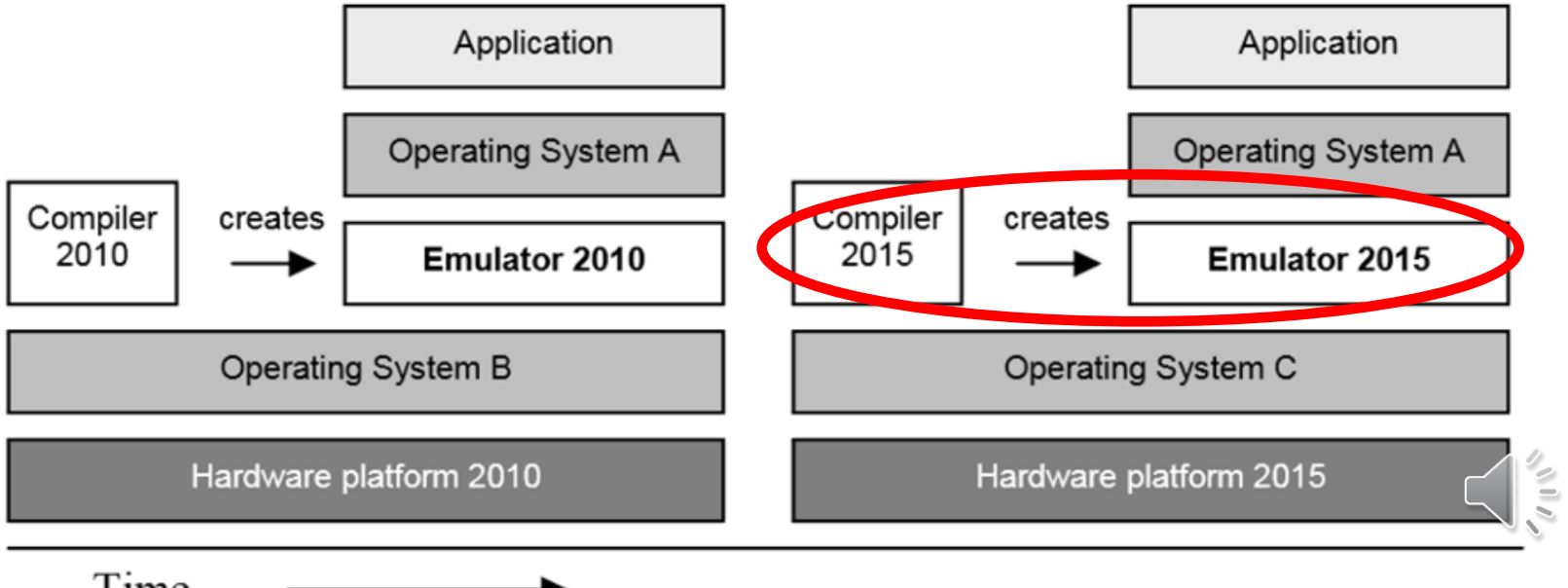
- Migration = recompiling to new guest
- Emulation chains = stacking on emulated guest
- (Universal) Virtual Machine = isolate from guest

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

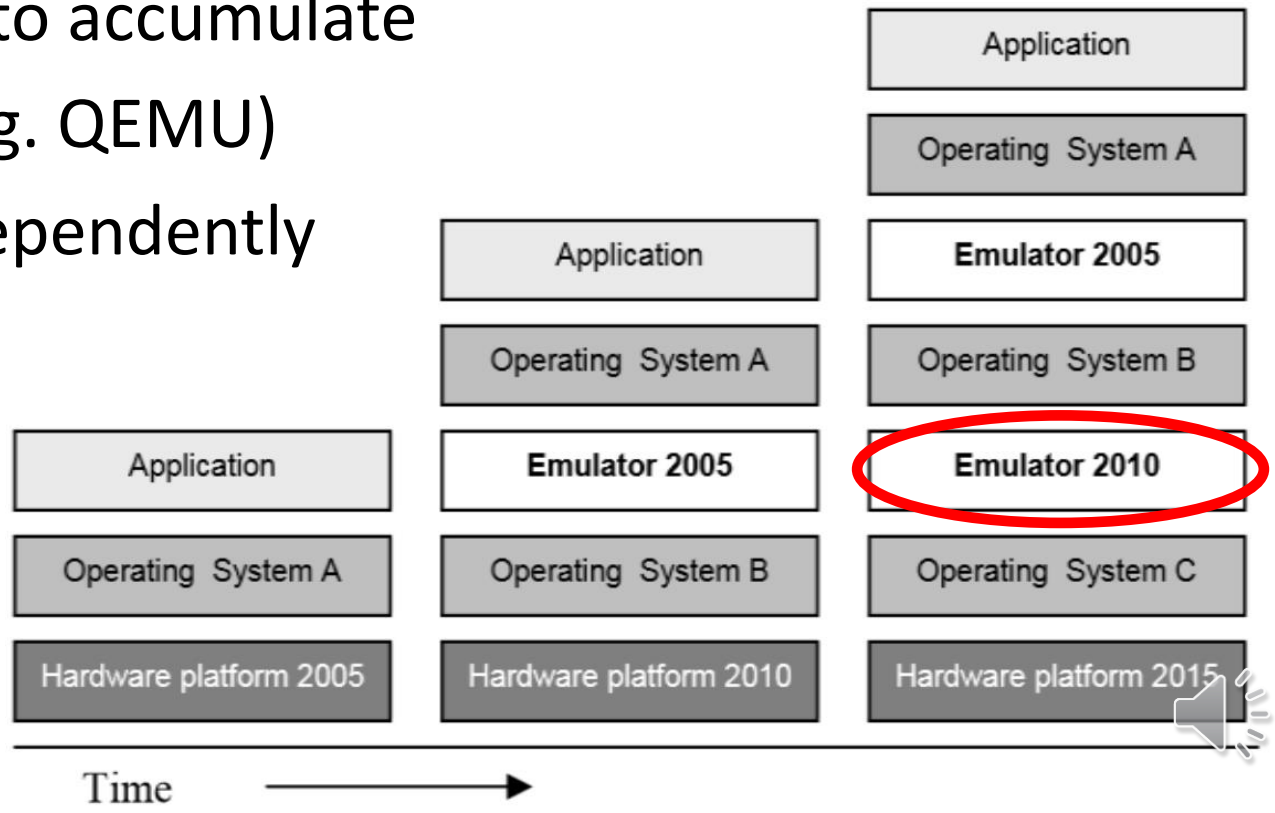


Emulator approach ➔ stacking emulators

IDEA: older system becomes emulated too

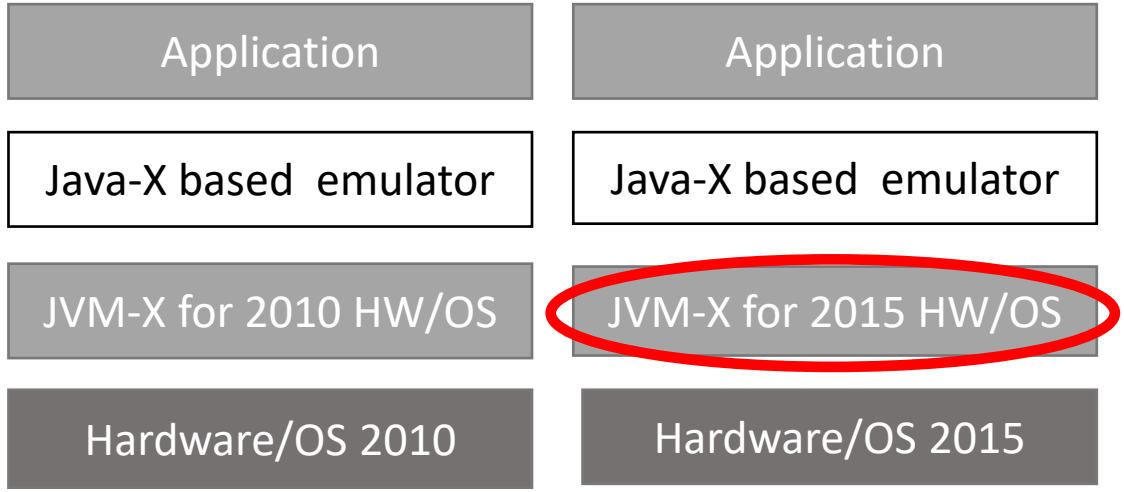
- Not same team working of different emulators (e.g. hidden win32 layer)
- Performance impact ➔ compensated by HW improvement
- Reliability impact ➔ more likely to accumulate
- Can be hidden/multiplatform (e.g. QEMU)
- Old emulator maintained (?) independently

- Alternative: virtual machine
  - If same CPU
  - Better performance

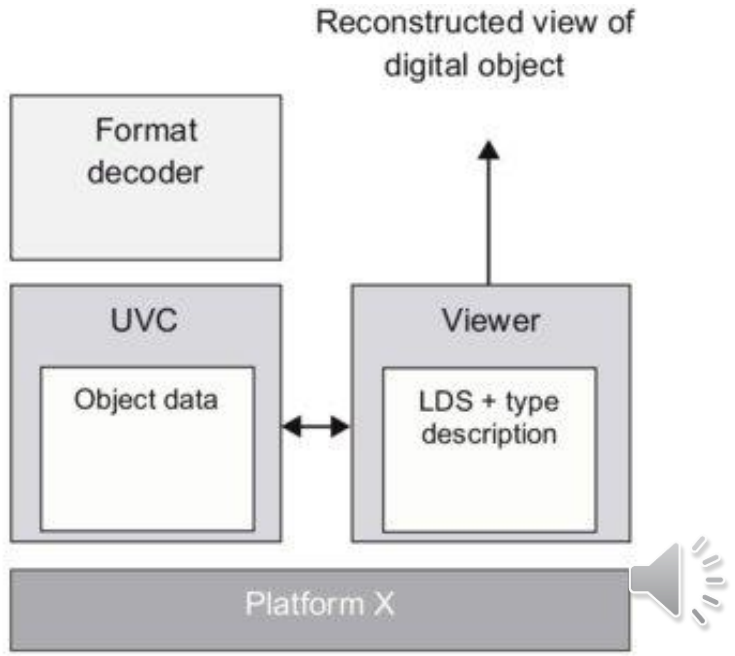


(Universal) Virtual Computer Approach

- 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
- Lee, Kyong-Ho, et al. "The state of the art and practice in digital preservation." Journal of research of the National institute of standards and technology 107.1, 2002
- Jeffrey van der Hoeven, Bram Lohman, Remco Verdegem, Emulation for Digital Preservation in Practice: The Results, The International Journal of Digital Curation, Issue 2, Volume 2, 2007
- Mark Guttenbrunner and Andreas Rauber. A Measurement Framework for Evaluating Emulators for Digital Preservation. ACM Trans. Inf. Syst. 30, 2, Article 14, May 2012
- Dirk von Suchodoletz, Requirements towards Emulation as a Long term Preservation Strategy, http://eprints.rclis.org/14860/ 2009
- 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://emulation.fandom.com/wiki/History_of_emulation
- https://kaluszka.com/vt/emulation/history.html
- https://www.zophar.net/articles/art_14-2.html
- https://blog.dshr.org/2016/07/talk-at-jisccni-workshop.html (blog of D. Rosenthal)