

Digital Preservation

Emulation

Mark Guttenbrunner
Institut für Softwaretechnik und Interaktive Systeme
TU Wien

<http://www.ifs.tuwien.ac.at/dp>

- What is Emulation?
- Concept and Definition
- Emulation in Digital Preservation
- Emulation View Path
- Preserving Emulators
- Emulation in Preservation Planning
 - With a short introduction to Preservation Planning

What is Emulation?

- *Emulation* refers to the capability of a device or software to replicate the behaviour of a different device or software
- examples: modem-emulation, terminal-emulation, emulation of computer-systems, video game system emulators
- difference to simulation ?
 - flight simulator does not actually fly

- obsolete programs
 - recompilation not possible because of missing source code
 - data cannot be migrated to different format (e.g. scientific analysis)
- multimedia (interactive art, video games)
 - the logic inside a program and the appearance have to be preserved
- preservation of software for historic reasons
 - e.g. early operating systems
- keeping documents authentic (e.g. electronic signatures)
- migration through emulation
- data archaeology

different levels of emulation:

- application (viewer)
- operating system (e.g. Wine)
- computer architecture
 - virtualization (e.g. DOSEMU on Linux)
 - hardware emulation (e.g. Dioscuri)
- interface level
 - output devices (e.g. mobile platforms vs. PC-screen)
 - input devices (e.g. paddle controls vs. mouse)
- environment (e.g. video game arcade in museum environment)

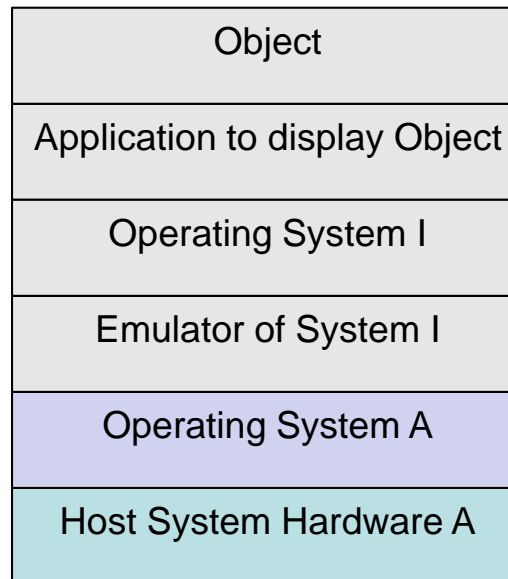
- virtual machine (virtualization)
 - only components are emulated, the code is run on the original CPU
 - can only be used, if the CPU of the host system is the same as the CPU of the target system.

- emulation of the full hardware of a system
 - all aspects of the original system are emulated

- techniques used
 - dynamic binary translation - to optimize speed
 - HLE (High Level Emulation) – to abstract e.g. video hardware to a functional level

Definition

- one possible definition of an emulator:
 - An emulator is a program that runs on one computer (the emulator's 'host' system) and thereby virtually recreates a different computer (the emulator's 'target' system).



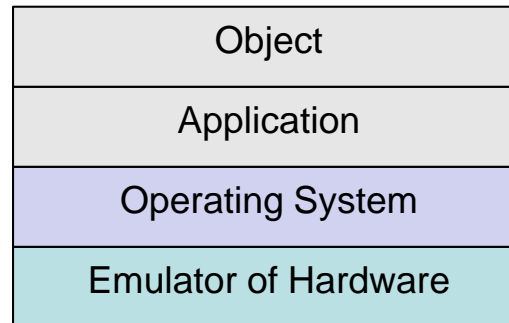
- emulation is usually done in software, emulation in hardware is also possible, but does not solve digital preservation problems!
- concept: keep the data in its original, unaltered form and keep using the software originally used to display the data
- the software has to be run on the operating system and the operating system on the hardware it was developed for

- context documentation of digital objects
- documentation about handling of objects
- not only technical but also social properties are relevant
- data transfer between emulated and host environment necessary (injecting digital objects into environment, extracting data from environment)
- stability more important than speed
- automation of processes (e.g. executing commands in target environment, automated input)

- expert knowledge necessary on how to use systems and programs (e.g. OS)
- not only the digital object, also all the necessary secondary digital objects and their settings have to be preserved (view path)
- emulators are programs that have to be preserved over a long term as well

Emulation View Path

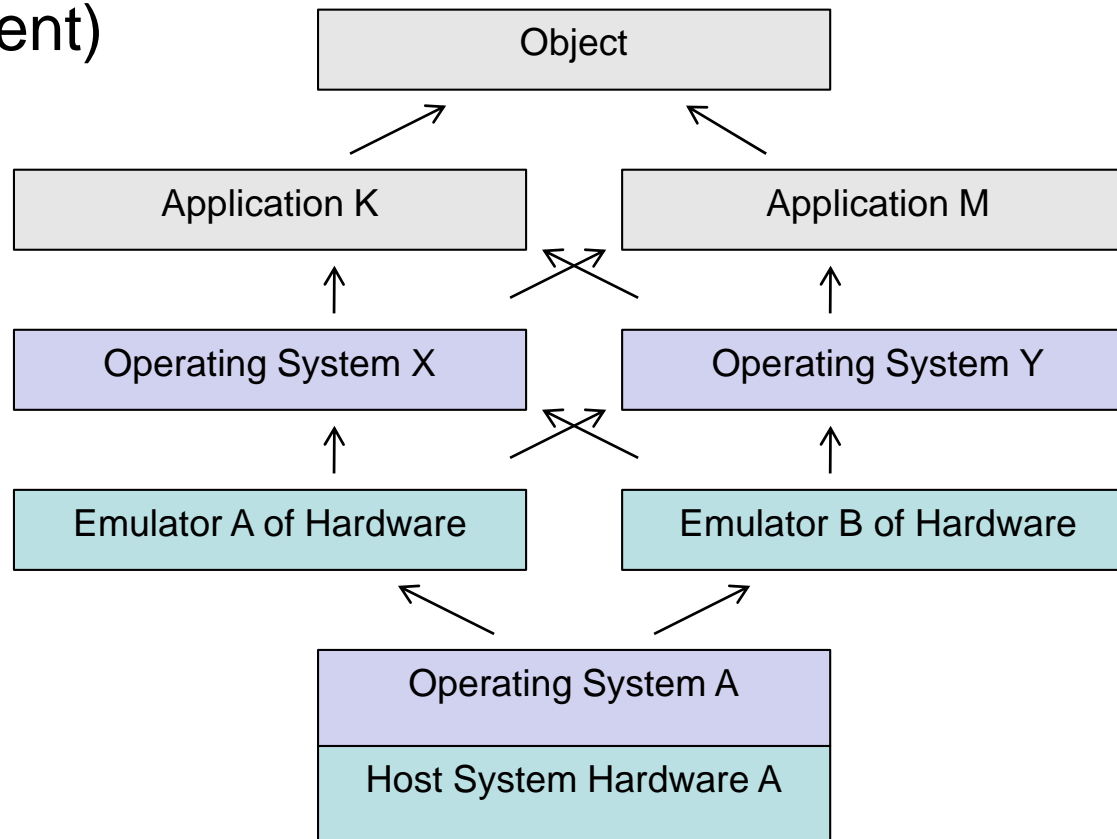
- View-Path: necessary path of secondary objects needed to display an object (e.g. application, operating system, hardware architecture)



- Example: Word for Windows 1.0 in Windows 3.0 on MS-Dos 5.0 using the DOSBox emulator on Windows Vista running on 64bit Hardware

Emulation View Path

- different view-paths are possible (different emulator, different OS, different application to display the same document)



- documentation of view path necessary
 - objects in the view path
 - settings of the objects

- storage of objects
 - with the object or separate
 - complete view path (e.g. disc-image) or discrete objects and configure on demand

- hardware configuration
 - CPU type, memory, configuration, speed settings
 - gfx-card, physics-card (3D rendering) incl. settings
 - sfx-card, settings
 - input-devices (e.g. light pen, trackball, data glove)
 - output-devices (type (vector/raster), aspect ratio, size, display settings like brightness, colours)
 - additional processing units, memory expansion cards (home-computers, game consoles)
 - depends on the original system, list is by far not complete!
 - tools can be used to determine the hardware and software settings (e.g. *Sigar*)

- operating system
 - type, version, system updates
 - font size
 - screen resolution
 - colour depth
 - installed fonts
 - appearance settings
 - installed utilities / applications that influence the appearance of the operating system

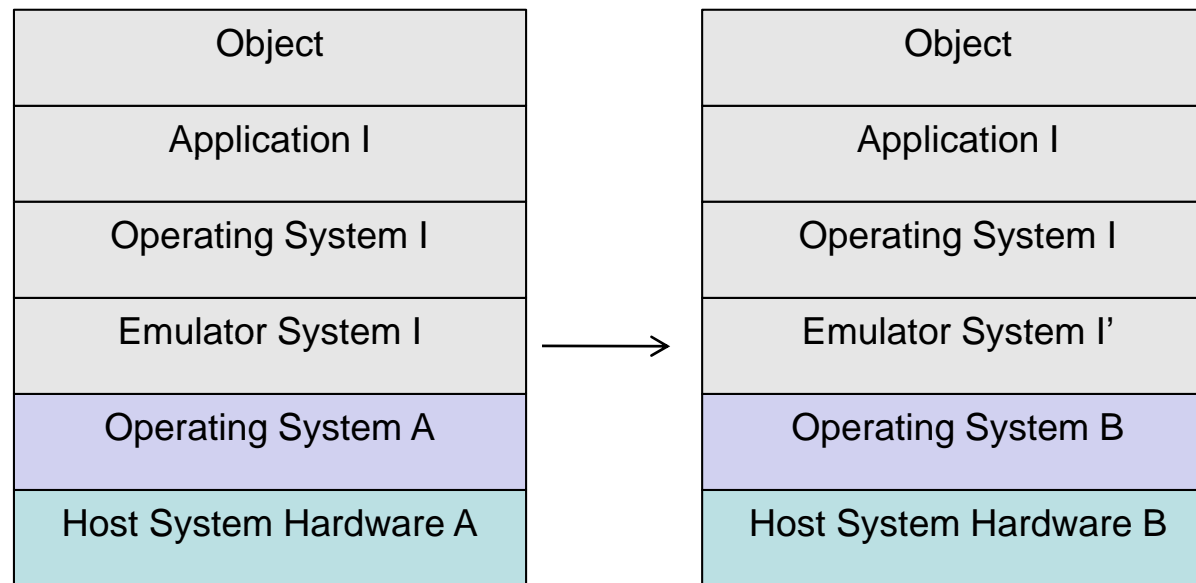
- needed secondary digital objects
 - virtual machine (e.g. JVM, .net)
 - database software (e.g. MySQL)
 - software device drivers (e.g. ODBC driver)
 - memory managers (e.g. dos4gw)
 - fonts
 - codecs
 - viewer/editor application (e.g. OpenOffice, PDF-Viewer)

- digital object itself
 - settings
 - configuration (e.g. appearance options, message boxes)

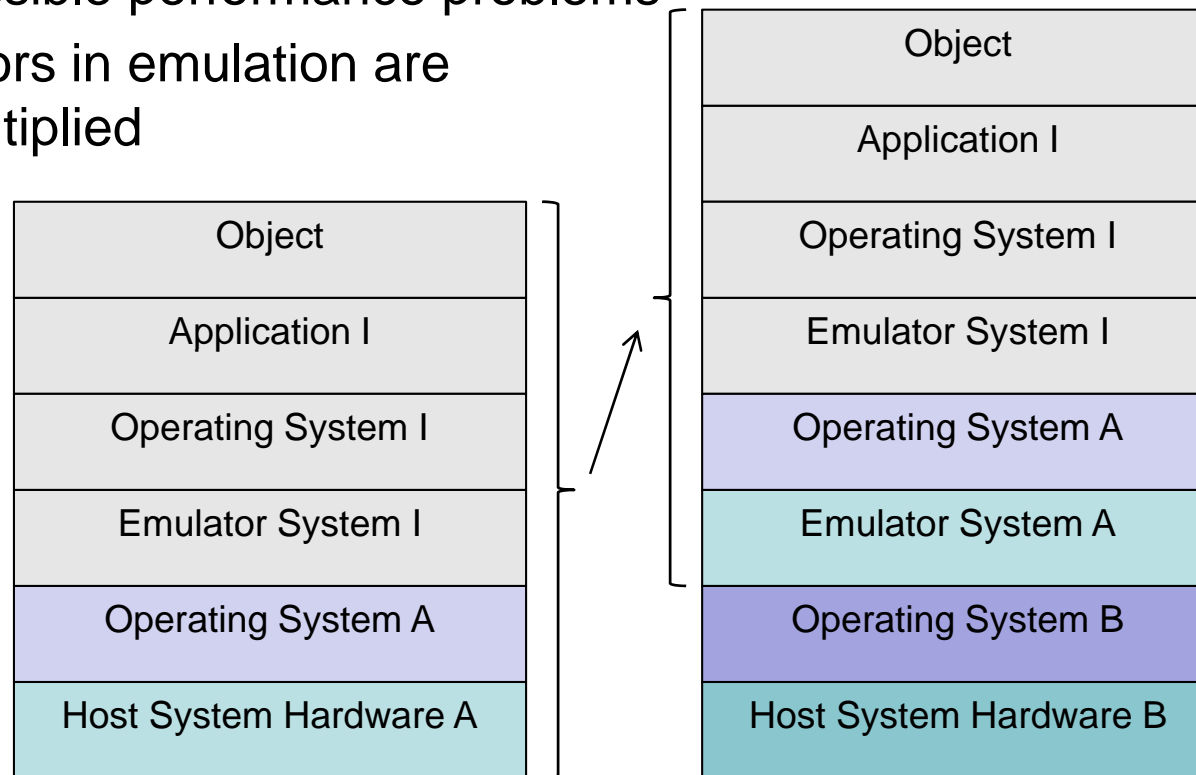
- additional objects not in the view path
 - can influence behaviour (e.g. speed)
 - e.g. virus scan software, remote desktop software

- Emulators get obsolete - software for specific platform (hardware or virtual machine)
- various strategies for preservation
 - re-hosting
 - stacked emulation
 - emulation virtual machine (EVM)

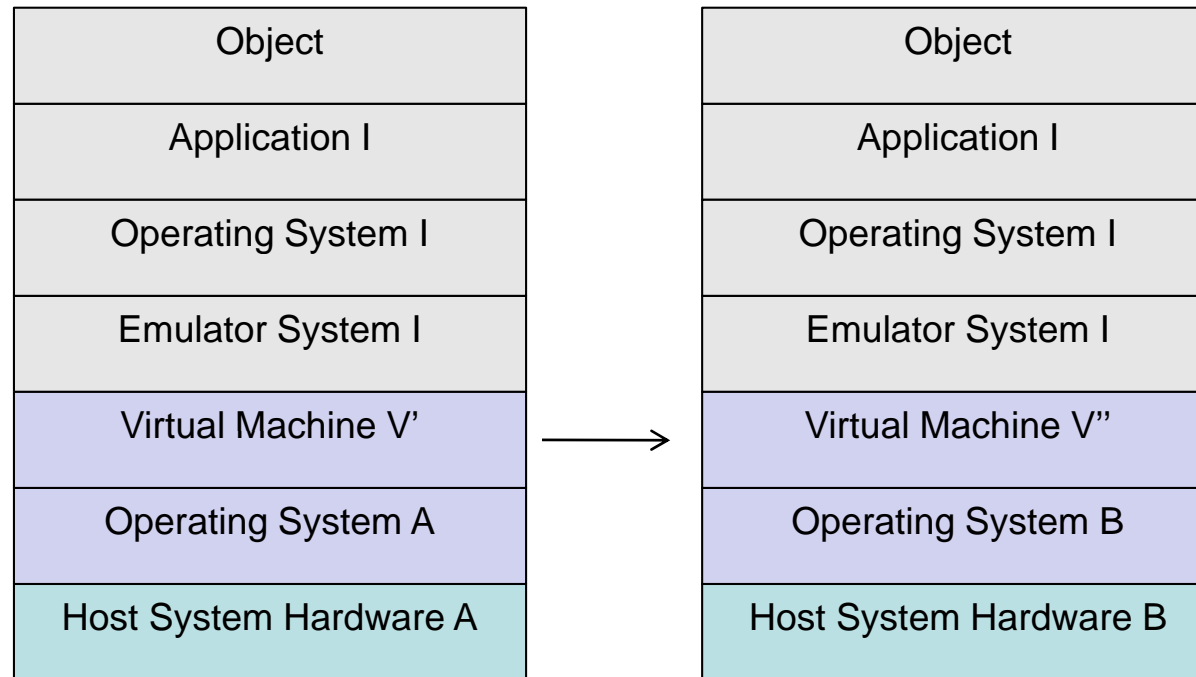
- re-hosting
 - migrating emulator to a different host system
 - has to be done for every emulator when the host gets obsolete
 - documentation of system and expert knowledge about its function has to be available at implementation time



- stacked emulation (*Russian Doll Syndrome*)
 - emulating the emulators host system
 - possible performance problems
 - errors in emulation are multiplied



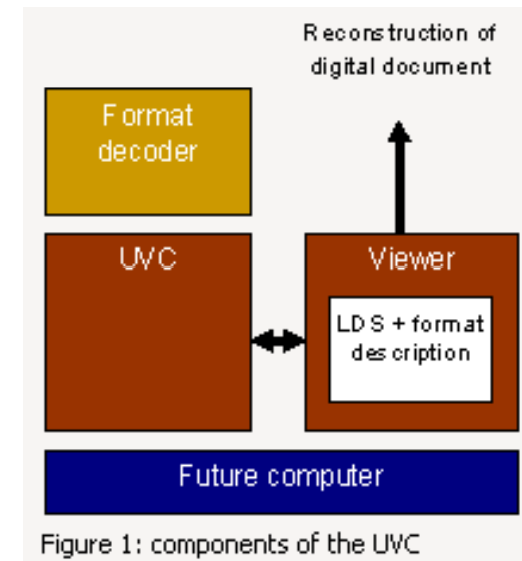
- emulation virtual machine
 - creating emulators for a virtual machine which is ported to a new host when the original host gets obsolete
 - only the virtual machine has to be ported to the new host



- component based development
- specific machines are built from the components through configuration
- some examples:
 - Dioscuri (Java, emulates x86 PCs)
<http://dioscuri.sourceforge.net/>
 - JPC (Java, emulates x86 PCs)
 - IronBabel (.NET, emulates modern video game consoles)
 - MAME/MESS (C, emulates video games, arcade machines, home computers)

UVC – Universal Virtual Computer

- developed by IBM / KB Netherlands for DP
<http://www.alphaworks.ibm.com/tech/uvc>
- not a “real” hardware platform
- simplified design, open specification -> virtual machine easy to implement on future host machine
- Components:
 - Universal Virtual Computer (UVC) (future)
 - Logical Data Schema (LDS) with type description (now)
 - UVC program (format decoder) (now)
 - Logical Data Viewer (future)
- mixed migration/emulation approach



Significant Properties

- properties of a digital object that are considered significant and as such have to be preserved
- Examples
 - image width, colour depth
 - page breaks, font, character encoding
 - relative speed
 - ...
- Preservation action should preserve the important significant properties
- importance of properties differ for institutional settings

Preservation Planning

- Consistent workflow leading to a preservation plan
- Analyses, which solution to adopt
- Considers
 - preservation policies
 - legal obligations
 - organizational and technical constraints
 - user requirements and preservation goals
 - Describes
 - the preservation context
 - evaluated preservation strategies
 - resulting decision including the reasoning
- Repeatable, solid evidence
- Requirements tree
 - preserving significant properties, tool, process, costs

Why Preservation Planning?

- Several preservation strategies developed
- For each strategy: several tools available
- For each tool: several parameter settings available
- How do you know which one is most suitable?
- What are the needs of your users? Now? In the future?
- Which aspects of an object do you want to preserve?
- What are the requirements?
- How to prove in 10, 20, 50, 100 years, that the decision was correct / acceptable at the time it was made?

- problems with dynamic and interactive content:
 - to get reproducible results the digital object has to follow a deterministic behaviour:
what are the factors that influence the objects behaviour?
 - continuous rendering of objects:
when should object properties be extracted?
where can properties be extracted from the running system?

deterministic behaviour:

- view path has to be constant to compare behaviour
- input has to be constant
 - macros
 - remote access
 - “hardware” (read input on hardware level on original system, apply on hardware-layer of emulator)
- external factors that influence deterministic behaviour have to be constant (e.g. date/time, network activity, random number seed)
- not every object’s behaviour can be made deterministic! (or not with justifiable effort)

how to extract significant properties:

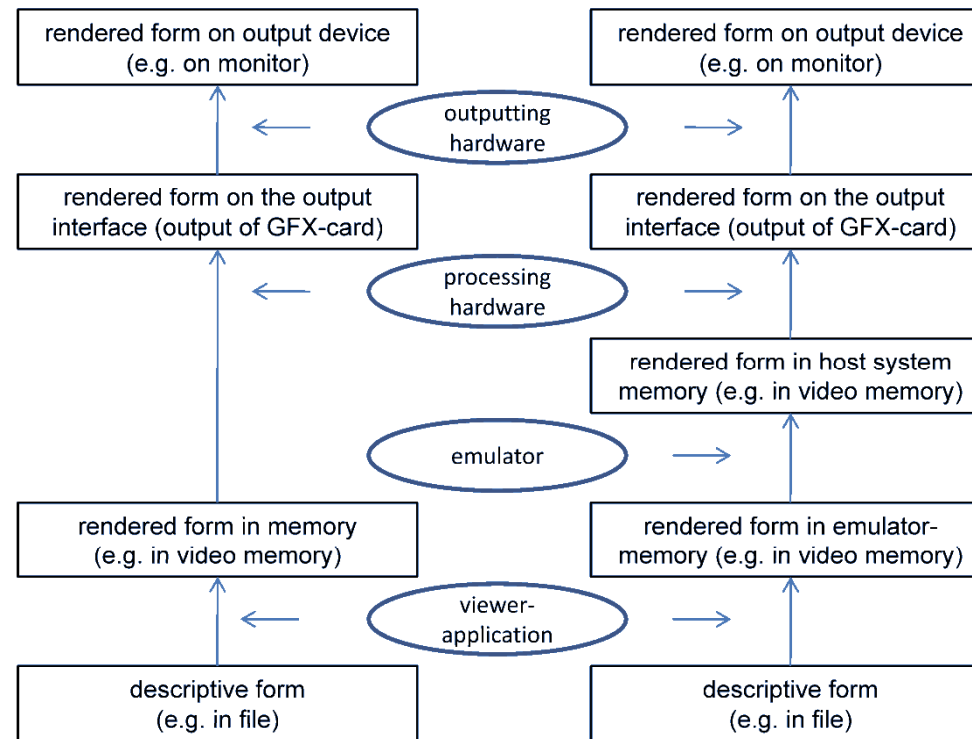
- not from the object, from the environment (object is rendered by the environment)
- environment has to support extraction ideally in extraction language (e.g. XCL) (not supported by emulators yet)
- properties have the dimension time (e.g. frames/second, cycles per second, number of file access operations per minute)
- properties change over time (e.g. frames/second min, average, max)

when to extract properties:

- not every state in an objects rendering process is significant
- depending on the object
 - target state: only one state after initially rendering the object or after applying a certain series of input events (e.g. rendering a static object)
 - series of states: only certain states after certain events (e.g. web site after each click on a link)
 - continuous stream: every rendered state of the object is important (e.g. video game, sound stream)

where to extract properties:

- rendered form of a digital object exists on various levels in a system:



where to extract properties:

- descriptive form
 - before rendering, useful for migration, no change in emulation
- rendered form in memory (original system)
 - both original system and emulator memory of host system
- rendered form in memory (host system)
 - exists only on host system, no comparison possible
- rendered form on output interface
 - signal analyzed by comparator (e.g. digital video signal, analog audio signal)
- rendered form on display device
 - influence of display device options (e.g. brightness settings)

where to extract properties:

- depending on the tests where to extract:
 - screenshot level after applying input: is the emulation working correctly ?
 - comparison of output: is the transformation from emulated system to host system working correctly ?
- depending on the original system:
 - screenshot only possible if not a single process system like video game device
 - output signal only possible if output device is not part of system (e.g. built-in speakers)
 - after output device always possible, additional factors influencing the signal

test workflow:

- determine external events that influence the objects behaviour
- describe the original environment
- decide on what level to compare the digital object
- recreate the environment in emulation
- apply standardized input to both environments
- extract significant properties
- compare the significant properties

Digital Preservation

Preserving Interactivity

Mark Guttenbrunner
Institut für Softwaretechnik und Interaktive Systeme
TU Wien

<http://www.ifs.tuwien.ac.at/dp>

- objects & motivation
- alternatives
- software preservation
- interactive art preservation
- video game preservation
 - case study for video game preservation
 - video game archives
 - video game preservation projects
- virtual world preservation

- documents
 - authenticity & accessibility
 - boundary document/application is fuzzy (e.g. scripts/macros embedded in documents)

- software
 - keep original proprietary software running and data accessible
 - distributed software is hard to preserve
 - scientific software for research
 - business software for legal obligations (safekeeping period)
 - computer museum (cultural heritage)

- interactive art
 - authentic look & feel
 - hardware proprietary
 - documentation not available
 - can be a mix of analogue and digital installations

- video games
 - same problems as interactive art (-> “art games”)
 - legal problems
 - proprietary media
 - companies are not supportive
(yet? -> Digital Game Preservation White Paper at Game Developers Conference 2009)
 - distributed games

- emulation
 - application level (documents)
 - OS level (software)
 - hardware (documents, software, interactive art, video games)

- migration
 - source ports (software, video games, interactive art)
 - static binary translation (software, video games, interactive art)
 - documentation on video, paper (documents, software, interactive art, video games)
 - database migration

- simulation
 - software
 - interactive art
 - video games

- reinterpretation
 - interactive art
 - video games

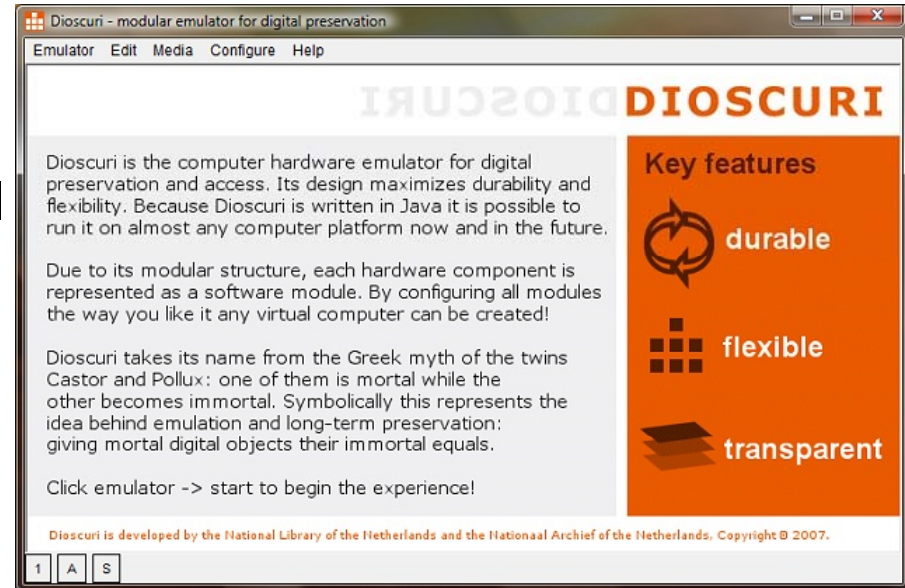
- mixed forms (e.g. reinterpretation/migration, reinterpretation/simulation)

- short term
 - “XP-Emulator” on Windows 7
 - DOS-compatibility-mode on Windows XP
 - Rosetta on Intel-based Apple Macintosh (dynamic binary translation)
- long term
 - hardware emulation
 - migration (reprogramming, porting) to new platforms
- JISC-study on significant properties of software
 - properties necessary of reconstruct software from source
http://www.jisc.ac.uk/media/documents/programmes/preservation/spsoftware_report_redacted.pdf

- Dioscuri

<http://dioscuri.sourceforge.net/>

- Intel x86 PC emulation
- emulator specific for digital preservation purposes
- features copy/paste from emulated environment
- stable solution (Java VM, modular emulation)



- Qemu

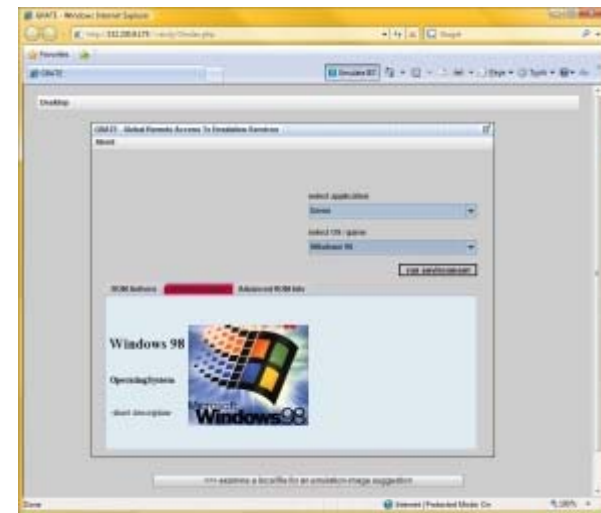
<http://www.nongnu.org/qemu/>

- emulation and (x86 only) virtualization
- different target CPUs supported (ARM, SPARC, PowerPC, MIPS etc.)

- Grate (Global Remote Access to Emulation-Services)

<http://planets.ruf.uni-freiburg.de/>

- TightVNC technology (Virtual Network Computing over web-access)
- different emulators supported
- transfer speed of input/output usually not fast enough for video games but sufficient for applications
- can be used for remote migration: transfer local file into emulated environment, migrate, transfer back to local system (without installing emulator locally)
- uses DROID / PRONOM services for recognizing file formats



- Archiving the Avant-Garde

<http://www.bampfa.berkeley.edu/about/avantgarde>

- Berkley Art Museum, Guggenheim museum
- the Variable Media Initiative (<http://www.variablemedia.net/>)
- Variable Media Questionnaire: artists choose different strategies for preserving art: emulation, migration and reinterpretation

- DOCAM (Documentation and Conservation of the Media Arts Heritage)

<http://www.docam.ca/>

- visual and performing arts (theatre, dance, performance) and architecture
- cataloguing structure, case studies in conservation and preservation, documentation and archival management

- Real-time Demos

- Amiga Demo-Scene Archive (ADA) (<http://ada.untergrund.net/>)
- Hornet Archive (PC Demos) (<http://www.hornet.org/>)

- “The Erl King” (1983-85) by Grahame Weinbren and Roberta Friedman
 - interactive movie, obsolete and generic hardware and software
 - original software was written by the artist -> very high priority to preserve the original code
 - emulated for Guggenheim museum
<http://www.bampfa.berkeley.edu/about/ErlKingReport.pdf>
 - migrated as webpage
<http://www.grahamweinbren.net/ErlKing/Erlkoenig.html>



hardware emulators (multiple platforms)

- MAME
Multiple Arcade Machine Emulator

<http://mamedev.org/>

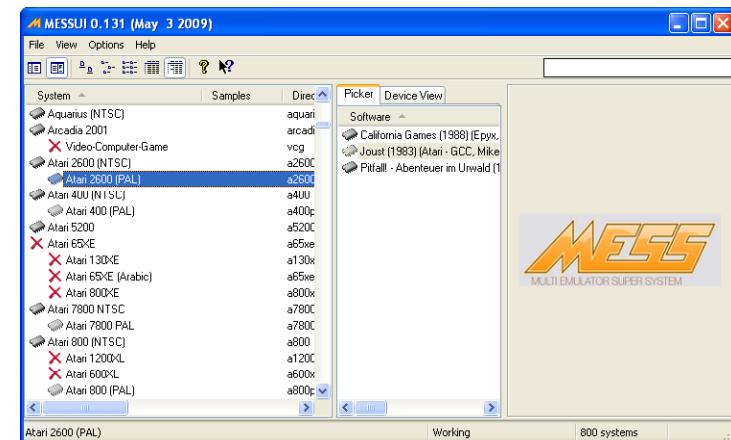
- arcade machines
- modular concept
- open source (C)



- MESS
Multiple Emulator Super System

<http://www.mess.org/>

- based on MAME source
- emulates home computers & video game consoles

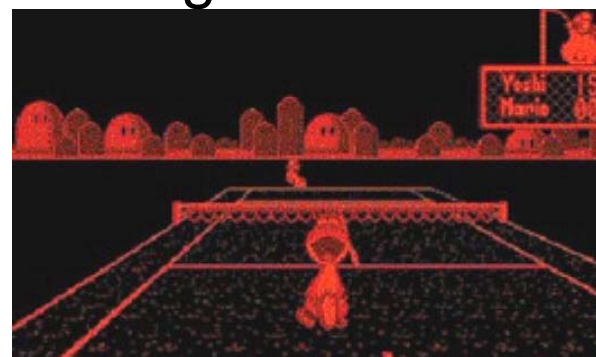


hardware emulators (single platforms)

- available for almost any system (from early home computers and video game console systems to Nintendo Wii)
- pro:
 - more feature complete emulation than multiple system emulators
 - better compatibility
 - more user friendly (less options, usually auto detection of some game settings like region)
- con:
 - typically platform dependent
 - usually single person development and not always open source

hardware emulators (single platforms) – some examples

- Atari 2600 - Stella
<http://stella.sourceforge.net/>
- Commodore Amiga – WinUAE
<http://www.winuae.de/englisch/main.html>
- Sony Playstation 2 – PCSX2
<http://www.pcsx2.net/>
- Nintendo DS – no\$cash
<http://nocash.emubase.de/gba.htm>
- Nintendo Virtual Boy – Red Dragon
<http://rdragon.vr32.de/>
- Sega Dreamcast – nullDC
<http://www.emudev.org/nullDC-new/>



game engine interpreters

- SCUMM-VM

<http://www.scummvm.org/>

- “Script Creation Utility for Maniac Mansion”
- various engines supported (e.g. LucasArts Games, Sierra)



- Frotz

<http://frotz.sourceforge.net/>

- Infocom Z-Machine
- text-adventures
- not true to original appearance

```
Bedroom 0/2

THE HITCHHIKER'S GUIDE TO THE GALAXY
Infocom interactive fiction - a science fiction story
Copyright (c) 1984 by Infocom, Inc. All rights reserved.
Release 59 / Serial number 851108

You wake up. The room is spinning very gently round your head. Or at
least it would be if you could see it which you can't.

It is pitch black.

>say "This isn't the game! It's a screenshot of the game!"
Talking to yourself is a sign of impending mental collapse.

>inventory
You have:
  a splitting headache
  no tea
```

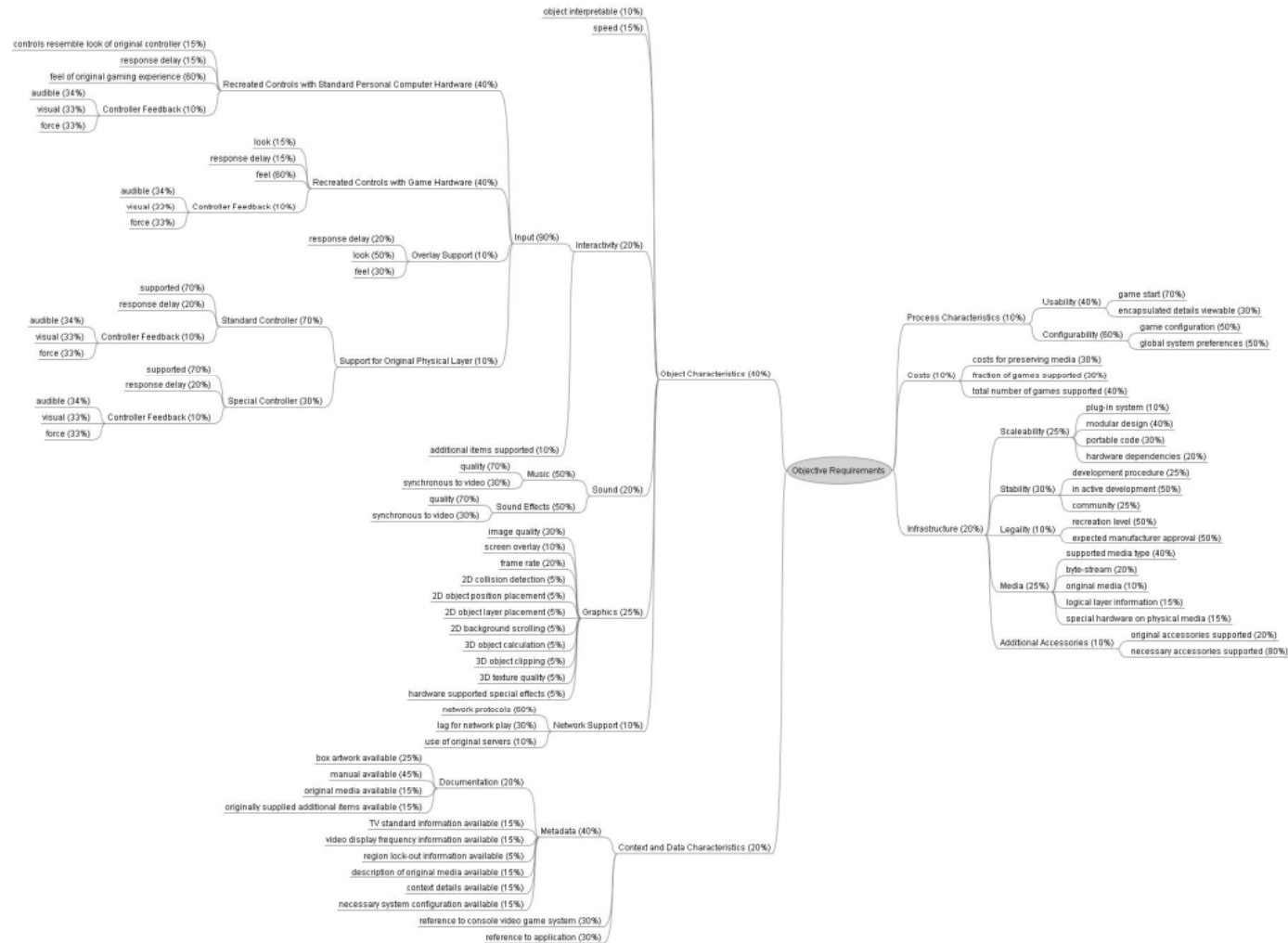
- cultural heritage
- public interest (“retro gaming”)
- problems:
 - broken hardware, decayed media
 - changed TV Standards
- challenges:
 - unavailable documentation (partially even lost forever)
 - preservation of game code from different media (optical, cartridge, online)
 - legal aspects
 - look & feel (different controllers)



- define basis for library scenario
 - data collection = console video games
 - designated user community are visitors of the library
 - required goals
 - authentic look & feel, long term preservation, accessibility and metadata, high compatibility, cost to be considered

- choosing sample records
 - three games for every system
 - popular game
 - game with special controller / overlay
 - game using special accessory or hardware enhancement or technically complex game

- identifying requirements



- considered alternatives
 - emulation
 - simulation
 - print-to-paper (video) approach

- developing the experiments
 - source code evaluation
 - game-play with keyboard/mouse and joystick
 - configuration of alternative

- running the experiments
 - alternatives for one console system
 - alternatives for different console systems from the same era
 - alternatives for one system of each era

- analyzing results
 - comparison of emulators of the one system and different systems of the same generation of video game consoles
 - games emulated well
 - special hardware only supported by dedicated emulators
 - scalability and stability better for multi-system emulators
 - no metadata supported
 - comparison of emulators of one system of each generation of video game consoles
 - infrastructure, process, costs and context similar results
 - emulation better for games of earlier systems
 - no perfect emulation for systems of the last three generations

- private collectors
 - very active community
 - <http://www.digitpress.com/>
 - <http://www.atariage.com/>
 - not relying on public funding
 - longest running initiatives
 - problem: lots of very rare specimen only in private hands, no consolidated preservation efforts

- Computerspielemuseum (Berlin, Germany)
 - <http://www.computerspielemuseum.de/>
 - huge collection from USK (Unterhaltungssoftware Selbstkontrolle)
 - relies mainly on donations for obsolete games/systems
 - exhibitions (e.g. Pong Story <http://pong-mythos.net/>)

- DiGA – Digital Games Archive
<http://www.digitalgamearchive.org>
 - raise public awareness concerning the cultural significance of entertainment software
 - guarantee its long term preservation
 - create a legal and dependable base for the preservation of games (legislative proposals etc.)
 - digitally donated games free for download
- National Video Game Archive (UK)
<http://www.nationalvideogamearchive.org/>
 - National Media Museum and Nottingham Trent University
 - “celebrate video game culture and preserve its history for researchers, developers, game fans and the public”
 - preserve, analyze and display the products of the global videogame industry

- Preserving Virtual Worlds (US) (started 2008)
<http://pvw.illinois.edu/pvw/>
 - preserving American art awards by Library of Congress US in 2007 - also for video games
 - two year project led by University of Illinois' Graduate School of Library and Information Science (GSLIS)
 - explores methods how to preserve digital games and interactive fiction
 - case studies on: Spacewar!, Second Life, Star Raiders, Doom, Warcraft

- KEEP - Keeping Emulation Environments Portable (EU-project) (started 2009)
<http://www.keep-project.eu/>
 - „Emulation Access Platform“ - tools for accessing and storing a wide range of digital objects
 - Disocuri, GRATE

- Extracting the world
 - user generated content
 - preserve interactivity with the world
 - convert to different format

- Recording events
 - interactions with the world
 - Interactions between users
 - “real-life” approach

- Second Life

- Virtual Conferences (Posters, Presentations)
- User generated art
- Events

http://www.archive.org/details/SL_AvatarIslandIntroduction

- World Of Warcraft

http://www.archive.org/details/Wow_ShattrathTour

- EA-Land (The Sims Online): The Final Countdown

- "Lost Server Connection"

http://www.archive.org/details/EALand_FinalCountdown

Digital Preservation

Practical Tasks

Mark Guttenbrunner
Institut für Softwaretechnik und Interaktive Systeme
TU Wien

<http://www.ifs.tuwien.ac.at/dp>

Practical Tasks

- tasks: <http://www.ifs.tuwien.ac.at/~guttenbr/teaching/dp2009/Emulation.zip>
 - *digital art*: run “Atari Robot” demo on Atari800 emulator
 - *video games*: run “Floppy Frenzy” in DOSBox at correct speed
 - *data access*: boot Dioscuri from the provided floppy image and copy text from the file “rogue\rogue.doc” into an email-message on your host-environment – also try to run the game Rogue
- to-do:
 - set up the emulation environment on your computer
 - run the application and execute the task
 - share your experience and observations with the class
- discuss these questions afterwards:
 - did you manage to set up the environment? where were the difficulties?
 - were you able to operate the system without problems?
 - can you tell if the experience was authentic? do you think it was? if not, why not?
 - what were the differences in the emulators? (if you ran more than one)

-
- 1. Präsentation nächste Woche
 - 4.5.2010 14:00-16:00
 - Seminarraum 188/2 (Favoritenstr., 4. Stock)

 - Fragen zu den Beispielen, zur Organisation ?

Thank you for your attention.

guttenbrunner@ifs.tuwien.ac.at
www.ifs.tuwien.ac.at/dp
www.planets-project.eu