

Digital Preservation: The State of the Art

Jeff Rothenberg

jeff@rand.org (310/393-0411, ext. 7703)

December 2002

Outline

- 1. What does “digital preservation” mean?**
- 2. Why is preserving digital documents difficult?**
- 3. Survey of proposed digital preservation approaches**

What does “permanent” mean?

“The ability of a material to remain stable over time and to resist chemical changes either from internal impurities or the surrounding environment.”

**—A Glossary for Archivists, Manuscript Curators, and
Records Managers (Society of American Archivists), p. 25.**

What does preservation mean?

“Enable reliable, authentic, meaningful and accessible records to be carried forward through time within and beyond organisational boundaries for as long as they are needed for the multiple purposes they serve.”

—Sue McKemmish

- **Preservation should allow future users to**
 - Retrieve, access, decipher, view, interpret, understand, appreciate, & experience
 - Informational artifacts (i.e., documents, data, records, etc.)
 - In whatever ways and for whatever purposes are desired in the future
 - While retaining their meaning & validity (i.e., “authenticity”)

Our documents *will* become increasingly digital

- For good, compelling reasons
 - Despite their vulnerability to loss

“Core digital attributes” :

- Digital documents can be copied perfectly
 - (Though this does not by itself guarantee longevity)
- Digital documents are easy to access and distribute
 - Due to the ease of communicating computer-readable documents
- Digital documents are easy to process
 - Search, modify, reformat, compute content
- Digital documents allow integrating and reusing information
 - From many sources
- Digital documents provide new *inherently digital* capabilities
 - To include networked, distributed hypermedia, etc.
- Digital documents save money and paper
 - Paperwork Reduction Act (U.S.)

And many new documents are *inherently digital*

- ***Inherently digital* documents are those whose meaning or usability arise from and rely on their being encoded in digital form**
- **They cannot be meaningfully represented as page images**
 - Doing so loses essential aspects of their contents and/or behavior
- **Examples include dynamic, active or interactive artifacts**
 - Multimedia (e.g., web pages, CD-ROM publications, Ph.D. dissertations)
 - Generated dynamically (e.g., calendars, agendas, bookkeeping data)
 - Generated on request (e.g., customized weather maps)
 - Generated automatically (e.g., JavaScript, cgi, ASP web pages, servlets)
 - Active presentation (e.g., animation, simulation, virtual reality)
 - Databases (where transactions update relationships and inferences)
 - Interactive (e.g., applets, interactive virtual reality)
- **Note: examples of these are difficult to show on static slides**
 - But I will show some examples that hint at the underlying problems

Choose What to Lose

- **We can “just save” original traditional documents**
 - Saving a document saves *all* of its attributes
 - Though some attributes may fade or decay over time
 - *Vernacular renditions* or “use-copies” are generated as needed from the original

- **But what is the equivalent for digital documents?**
 - Any approach to digital preservation entails choices—whether explicit or implicit
 - Of which attributes to save (and therefore which ones to sacrifice)
 - Ideally want *both* originals *and* vernacular renditions for future processibility

Saving “originals”

- **An “original” traditional document is well defined**
 - It is a distinct, physical artifact
 - Saving it saves all attributes of the document
 - Saving it is well-defined and straightforward

- **Can we define an equivalent concept for digital documents?**

A “digital-original” is any representation of a digital document that has the maximum likelihood of retaining all meaningful and relevant aspects of the document.

- Note that we must abandon physicality and uniqueness
- What kind of thing would this be?
- Can we create such a thing?
- How would we save such a thing?

Need to define *Preservation Principles & Criteria*

Alternative relationships between a preserved digital document and its original:

- **Same description**
 - Abstraction of attributes
 - Preserve nothing but descriptions, i.e., metadata
- **Same “content”**
 - Semantic attributes
- **Same “look-and-feel”**
 - Behavioral attributes
- **Same functionality and relationships to other artifacts**
 - Functional attributes
- **Same for all intents and purposes**
 - Analogous to an original traditional artifact

Possible Preservation Principles

- **For archives**
 - Enable future users to understand the roles that records *originally* played
 - In the business processes of the organizations that *originally* generated & used them
 - And they should be able to continue to use the records in future business processes
 - Which may require them, e.g., to determine past accountability
- **For deposit libraries**
 - A preserved publication should be as much like its *original*, published form as possible
 - Retaining its *original* behavior, functionality, and look-and-feel, as well as contents
 - It should not constitute a “reformatting” or “republication” of the *original*
- **For data warehouses**
 - Future users should be able to explore all implicit relationships in the *original* data
 - Even if original users were unable to see (or define) some of those relationships
- **For museums**
 - Future users should experience works of art just as they were *originally* experienced
 - Does this mean they should be as they were *originally*, or should they be reinterpreted?
- ***Derive specific, testable Preservation Criteria from the chosen principle***

But digital documents are very vulnerable to loss

- **Media decay or “evaporation” of bits**
 - Due to physical, chemical, magnetic effects, etc.
- **Media obsolescence**
 - Physical and logical format incompatibilities
 - Unavailability of suitable “drives” or “controllers”
- **Dependence on incompatible or obsolete software**
 - e.g., for word processing or hypermedia documents, DBs, etc.
- **Dependence on obsolete software environments**
 - Unavailability of OS, I/O drivers, etc. for required software
- **Dependence on obsolete hardware**
 - Unavailability of hardware required to run required software

Solving the media problem is “straightforward”

- **Truly “archival” digital storage media are not yet cost-effective**
 - **Since media (and their formats & reading devices) become obsolete so fast**
 - **And storage capacity, density, & speed increase with each new generation**
 - **The market will not pay for long-lived media while this progression continues**
- **So, must copy documents to new media while still readable**
 - **The same as for non-digital documents**
 - **However, must take into account obsolescence as well as physical lifetime**

But all digital documents are software-dependent

- **Digital documents can be seen only by running a program**
 - They are stored in encoded form, understood only by a program
 - They cannot be accessed, read, or printed without that program
 - They must be interpreted to be made intelligible to a human
 - They are essentially programs
 - Examples: ASCII character stream, hypermedia, database, animated film, interactive video game
- **The data file for a software-dependent document is *not* enough**
 - The file can be properly interpreted only by its software
 - Without the software, the document is unusable (may not even really exist)
 - “Virtual documents” may consist of multiple (distributed) files
- **Software-dependent documents are really *system*-dependent**
 - They require a software environment (OS, drivers, etc.)
 - Which in turn requires a hardware environment (CPU, I/O devices, etc.)

What makes a digital artifact software-dependent?

- **It is meaningful only to its original software**
 - Its structure & content can be understood only by the program that created it
 - We can make sense out of it only by running this software
- ***Or* we need to see exactly what its author or reader saw**
 - To understand the source of their insights or blind-spots (e.g., for research)
 - To hold them legally or ethically accountable for what they should have been able to infer from the artifact
 - To understand their appreciation of the artifact
 - To experience some subtle artistic effect
 - i.e., when an artifact must retain its original functionality

A particular “view” of information may be crucial

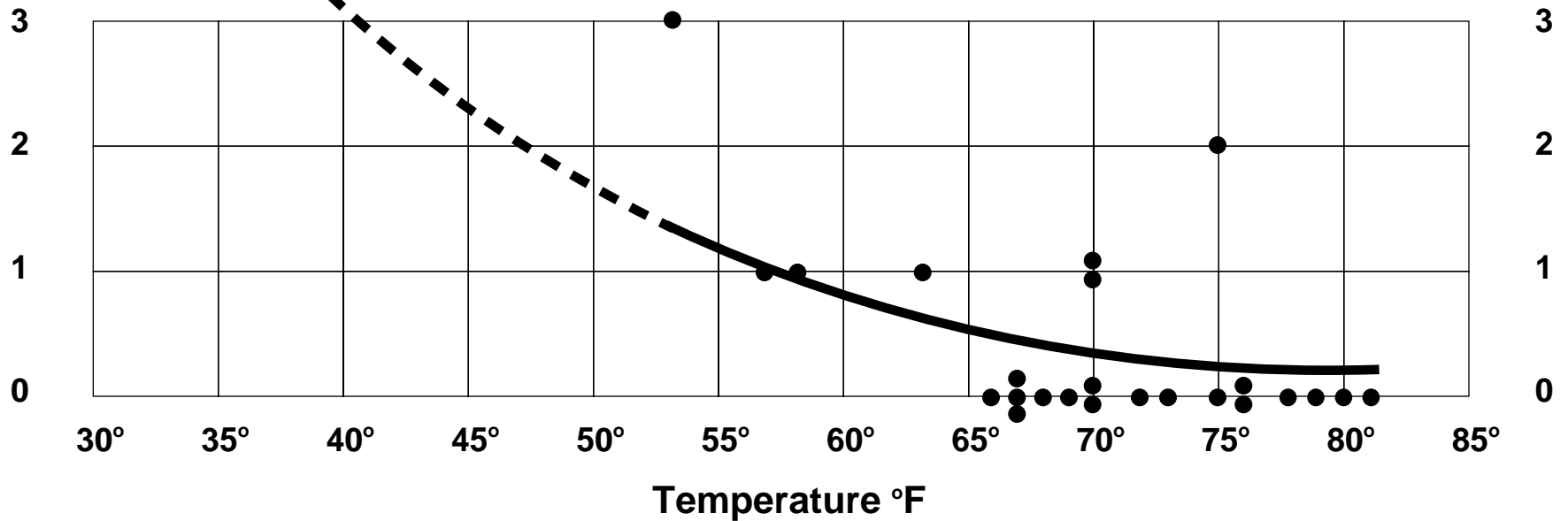
**Example: Space Shuttle O-ring damage vs. temperature
Prior to the Challenger disaster**

| | | | | | | | | | | | | | | | | | | |
|--|----------|-----------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Levels of O-ring damage | 3 | 1 | | | | | | | | | | | | | | | | |
| | 2 | | | | | | | | | | | 1 | | | | | | |
| | 1 | | 1 | 1 | 1 | | | | 2 | | | | | | | | | |
| | 0 | | | | | 1 | 3 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| | | 53 | 57 | 58 | 63 | 66 | 67 | 68 | 69 | 70 | 72 | 73 | 75 | 76 | 78 | 79 | 80 | 81 |
| | | Temperature °F | | | | | | | | | | | | | | | | |

Revealing view of Space Shuttle O-ring Data

Extrapolation of damage curve to the 31° F temperature forecast for Challenger's launch on January 28, 1986.

Dots indicate temperature and O-ring damage for 24 successful launches prior to Challenger. Curve shows that increasing damage is related to cooler temperature.



Even seemingly innocent changes...

The Periodic Table of the Elements

| | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| H | | | | | | | | | | | | | | | | | He |
| Li | Be | | | | | | | | | | B | C | N | O | F | Ne | |
| Na | Mg | | | | | | | | | | Al | Si | P | S | Cl | Ar | |
| K | Ca | Sc | Ti | V | Cr | Mn | Fe | Co | Ni | Cu | Zn | Ga | Ge | As | Se | Br | Kr |
| Rb | Sr | Y | Zr | Nb | Mo | Tc | Ru | Rh | Pd | Ag | Cd | In | Sn | Sb | Te | I | Xe |
| Cs | Ba | * | Hf | Ta | W | Re | Os | Ir | Pt | Au | Hg | Tl | Pb | Bi | Po | At | Rn |
| Fr | Ra | ** | Rf | Ha | Sg | Ns | Hs | Mt | | | | | | | | | |

* La Ce Pr Nd Pm Sm Eu Gd Tb Dy Ho Er Tm Yb Lu

** Ac Th Pa U Np Pu Am Cm Bk Cf Es Fm Md No Lr

...can lose information

H He
Li Be B C N O F Ne
Na Mg Al Si P S Cl Ar
K Ca Sc Ti V Cr Mn Fe Co Ni Cu Zn Ga Ge As Se Br Kr
Rb Sr Y Zr Nb Mo Tc Ru Rh Pd Ag Cd In Sn Sb Te I Xe
Cs Ba * Hf Ta W Re Os Ir Pt Au Hg Tl Pb Bi Po At Rn
Fr Ra ** Rf Ha Sg Ns Hs Mt

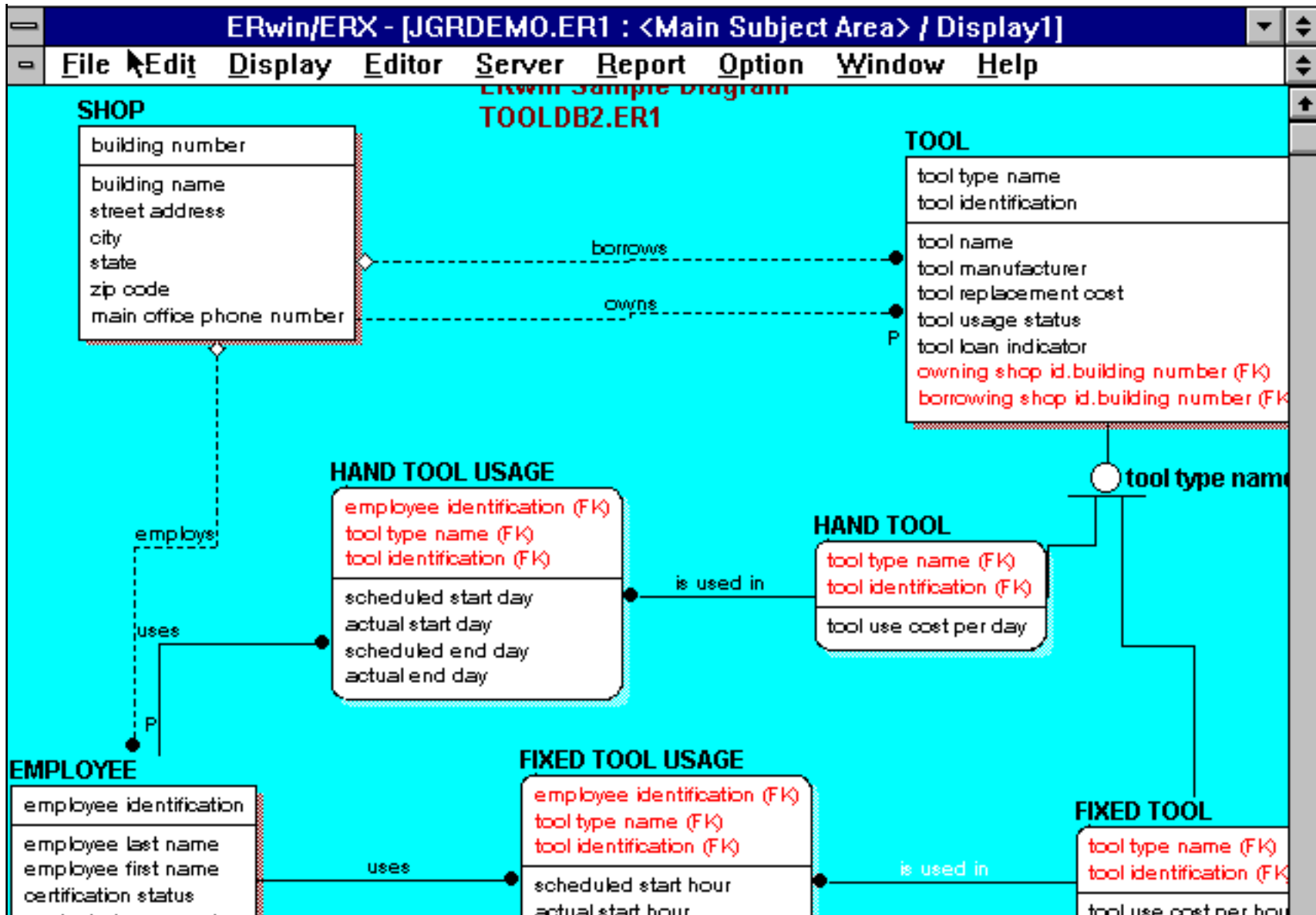
* La Ce Pr Nd Pm Sm Eu Gd Tb Dy Ho Er Tm Yb Lu

** Ac Th Pa U Np Pu Am Cm Bk Cf Es Fm Md No Lr

“to argue that any software or digital format is necessary to preserve the periodic table is patently absurd.”

—Ken Thibodeau “Challenges in Coming Years” in *The State of Digital Preservation: An International Perspective*, CLIR, July 2002

What you see may *not* be what you get



Text may not tell the story at all

V2.24 ERwin

```
if
  %JoinPKPK(oldrows,newrows," <> "," or ")
then
  select count(*) into numrows
  from %Child
  where
    %JoinFKPK(%Child,oldrows," = "," and");
  if (numrows > 0)
  then
    signal parent_updrstrct_err
  end if;
end if;
if
  %JoinPKPK(oldrows,newrows," <> "," or ")
then
  update %Child
  set
    %JoinFKPK(%Child,newrows," = "," ,")
  where
    %JoinFKPK(%Child,oldrows," = "," and");
end if;
```

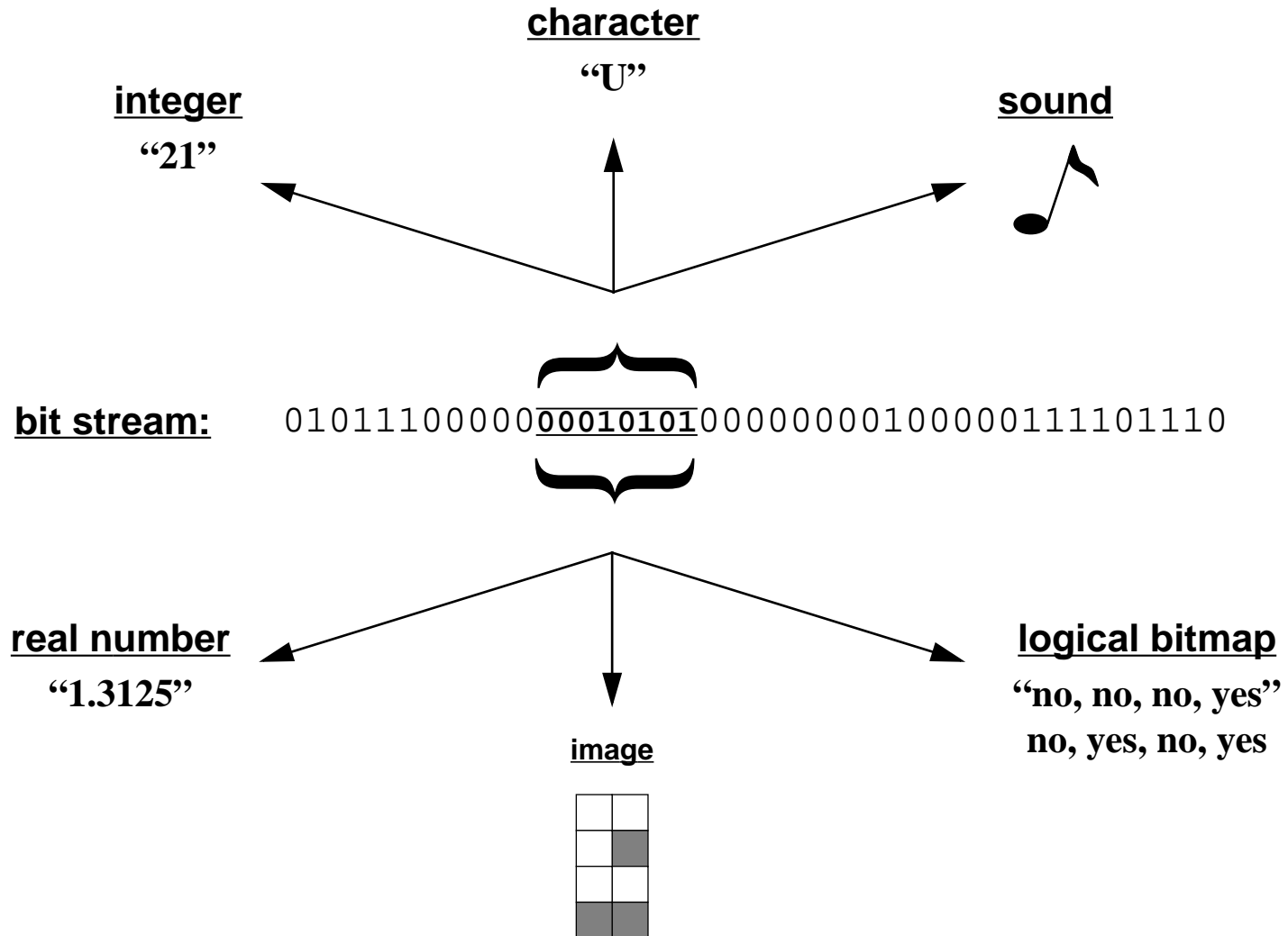
Every digital document is really a program

- ***A program***
 - Is a sequence of commands in some formal language
 - That is intended to be interpreted
 - By an interpreter that understands that language
- ***An interpreter***
 - Is an active process
 - That knows how to perform commands
 - Specified in a given formal language
- **Interpretation ultimately involves hardware**
 - ASCII codes are rendered by a printer or display
 - More complex entities are interpreted by software (applications)
 - But all S/W is ultimately interpreted by hardware

Interpreters can be hardware or software

- **Hardware interpreters are limited**
 - Can interpret only simple, static languages
 - And hardware *must* be well-specified in order to be built
- **Software interpreters are more flexible**
 - Can interpret more complex, dynamic languages
 - And software *need not* be well-specified in order to run
- **So most digital artifacts rely on software interpreters**
 - E.g., application programs
 - Which in turn may depend on other software interpreters
- **But software must ultimately execute on hardware**
 - So any chain of software interpreters must end in a hardware interpreter

Bits in a bit stream can represent *anything*



Saving the bits is necessary but not sufficient

- **Saving the bit stream of a document without saving its interpreter**
 - Is like saving hieroglyphics without saving a Rosetta Stone
- **But *worse*, since an interpreter is not just another document**
 - It is software
 - Which must be executed (i.e., interpreted)
 - And the document must still be understood (i.e., further “interpreted”)
- **So digital documents are generally software-dependent**
 - And software is ultimately hardware dependent

Digital vs. traditional informational artifacts

- **Traditional informational artifacts**
 - Require only final, human interpretation
 - But no prior interpretation to render them

- **Digital informational artifacts**
 - Require prior interpretation to render them (make them “visible”)
 - Plus final, human interpretation

Why is it hard to preserve digital documents?

- **Can't “just save” digital documents like physical documents**
 - The medium carries all attributes of a traditional document
- **Digital documents require an extra “interpretation” step**
 - To be made human-readable
 - Especially if they are dynamic, responsive, interactive, or “active” (executable)
 - But even simple text formats require interpretation
- **An interpreter can be hardware or software**
 - But hardware is limited to interpreting simple, static languages
 - And must be well-specified in order to be built
 - Whereas software can interpret more complex, dynamic languages
 - And need not be well-specified to run
- **So most digital documents rely on software interpreters**
 - E.g., application programs
 - Which in turn may depend on other software interpreters
- **But “execution” ultimately requires hardware**
 - So any chain of software interpreters must end in a hardware interpreter

Evaluating proposed preservation approaches

- **Capabilities**
 - What can it do? (assuming it can be made to work!)
- **Limitations**
 - What can't it do? (even if it worked perfectly)
- **Scalability**
 - To large numbers of documents of different kinds in different contexts
- **Probable costs and complexities**
 - Cost per-document, per-format, per platform
 - Including metadata requirements
- **Likelihood of success**
 - How feasibility is it?
 - Is it likely to be practical on a large scale over long time periods?

Overview of proposed approaches to preservation

- **Non-solutions**
 - Do nothing
 - Digital archaeology
- **Partial solutions**
 - Save page-images of documents
 - Extract and save “core contents” of documents
 - Translate documents into standard or “canonical” forms (without migration)
 - Rely on “viewer” programs to render obsolete formats in the future
 - Save metadata to help interpret saved bit streams (“assisted archaeology”)
 - Save source-code of rendering software (for future reverse-engineering)
- **Potentially complete solutions**
 - Formalization (replace documents by formal descriptions of themselves)
 - Migration (repeatedly convert documents into new formats)
 - Emulation (run original rendering software on virtually recreated hardware)

Non-solutions

- **Do nothing**
 - Let inactive obsolete digital documents disappear
 - Documents that remain active will be migrated by necessity
 - Others are not worth the effort
 - *But this decision cannot be undone once documents become obsolete*
- **Digital archaeology**
 - Let future archivists (and historians, scholars, etc.) worry about the problem
 - Rely on “cryptographic” techniques to decipher unintelligible bit streams
 - Since lots of cheap computing power will be available in the future
 - *But bit streams are inherently uninterpretable without additional information*

Partial solutions

- **Save page-images of documents**
 - In non-digital form (printing, microfilming, engraving, etc.)
But this sacrifices the documents' core digital attributes
 - In digital form (TIFF, PDF, JPEG, etc.)
But then still have the problem of preserving these digital formats
 - *And in any case this cannot preserve inherently digital documents*
- **Extract and save “core contents” of documents**
 - As text, data, etc. (using data-extraction programs, semantic XML tags, etc.)
 - *But then must decide—in advance—what constitutes their core contents*
- **Translate documents into standard or “canonical” forms**
 - Rely on such forms to last forever, without the need for migration (i.e., conversion)
 - *But must convert any non-conforming originals up-front to do this*
 - *And there are so many different standards to choose from*

Problems with standards

- **Ultimate standards are not realistic in the foreseeable future**
 - Information science is still inventing itself
 - Even the categories of kinds of information processing are not yet clear
 - So ultimate standardization is premature
- **Using successive, evolving standards would require translation**
 - But translation between standards is rarely reversible without loss
 - So this cannot reconstruct an original artifact
 - Translation forward across “paradigm shifts” may be impossible
 - So old artifacts may eventually be abandoned or corrupted
- **Evolving standards will always lag behind state-of-the-art use**
 - Until information science stops evolving
 - So state-of-the-art artifacts are likely to be “orphaned”
- **Can’t force users to conform to constraining standards**
 - This asks them to forego the use of new capabilities
 - Which are the motivation for using information technology in the first place

More partial solutions

- **Rely on “viewer” programs to render obsolete formats in the future**
 - Instead of the original rendering programs for each format
 - These can be “readers” rather than “editing” programs
 - And each viewer can render many different formats (e.g., GraphicConverter)
 - *But unlikely to be perfect unless provided by original vendors (e.g., Acrobat Reader)*
 - *Support for obsolete formats will eventually be discontinued*
 - *Viewer programs themselves become obsolete, so must port or rewrite them to run on future machines—and every port or rewrite incurs the risk of new bugs*
- **Save metadata to facilitate interpretation of saved bit streams**
 - Can think of this as “assisted archaeology”
 - Several such schemes have been proposed
 - *But this is even harder than formalization*
 - *And must decide—in advance—what are the core aspects of saved formats*
- **Save rendering software source-code (for future reverse-engineering)**
 - Another form of “assisted archaeology” but saving software instead of “metadata”
 - Source-code provides the semantics of the rendering software
 - So need not rely on being able to run obsolete “object code” on future computers
 - *But understanding & reverse-engineering source-code is notoriously difficult*
 - *And the behavior of resulting reverse-engineered programs would be hard to verify*

Potentially complete solutions

- **Formalization**
 - Replace documents by formal descriptions of themselves
 - Must understand the semantics of *every format and every document!*
 - Must guess what future readers will care about
- **Migration**
 - To be discussed subsequently
- **Emulation**
 - To be discussed subsequently

Formalization is very difficult

- **A formal description of a saved digital artifact's logical format**
 - Would allow properly interpreting that format in the future
 - So long as the formal description itself remained understandable
 - This would allow properly rendering the saved digital artifact
 - Without running its original software

- **Unfortunately computer science cannot do this very well yet**
 - Even for well-documented, well-defined formats
 - Let alone for arbitrary, new, proprietary formats

- **The only complete description of a format is its interpreter**
 - i.e., the software that knows how to render it

The promise of standards and migration

- **Standards can keep digital documents readable**
 - So long as documents conform to standards
 - *And* so long as those standards remain in common use
- **When standards evolve or become obsolete, rely on “migration”**
 - That is, convert (translate) documents into new forms as necessary
 - Ideally converting into new standard forms
- **Migration has a long history in computer science**
 - Programs & their data (as well as documents) have been translated as necessary
 - Though it is expensive and labor-intensive
 - And it is not always possible, feasible or cost-effective

Limitations of migration

- **Migration is a strategy based on faith**
 - No one can say what migration will require—or even if it will be possible!
 - No one can predict how much it will lose
 - It requires repeated conversion

- **Migration must be applied *repeatedly* to every individual document**
 - Despite the fact that most documents are rarely accessed
 - Each format, application, and document-type will require special handling
 - Non-standard documents or types of documents may be lost or corrupted

- **Migration may be infeasible across paradigm shifts**
 - Example: hierarchical -> relational DB conversion was not feasible (required redesign)
 - The first version of any new format is always the least powerful, i.e., “worst first”
(Example: translating HyperCard into HTML before XML existed)

Potential solution: run *original* software

- **To interpret a digital document's original format**
 - Which must be preserved as a bit stream
- **Not necessarily the software that created the document**
 - But software that understands how to “render” it
- **Saving original software is straightforward**
 - Just save the appropriate bit streams
 - Though this includes operating system environments as well as applications
- **But obsolete software will require obsolete hardware**
 - So must either save old computers in a “runnable” state
 - Or recreate them virtually in the future, i.e., using emulation

Saving obsolete hardware is infeasible

- **It sounds romantic but is not realistic**
 - As generations of hardware/software multiply, this becomes a nightmare
 - How long can we keep old hardware running cost-effectively?
- **It would greatly restrict access to old documents**
 - To a few “computer museums”
 - Losing the distributed aspect of digital information
- **It ignores storage media problems**
 - Neither old documents nor old software will last on their original media!
 - Once copied onto new media, they won’t fit in their original machines
 - So would need to build and maintain a new hardware interface to each old machine for each new generation of storage media
- **This is best reserved for “heroic efforts”**
 - Retrieving information from old media that may still be readable
 - Verifying the behavior of emulators for old systems and environments

So emulate obsolete computers on future computers

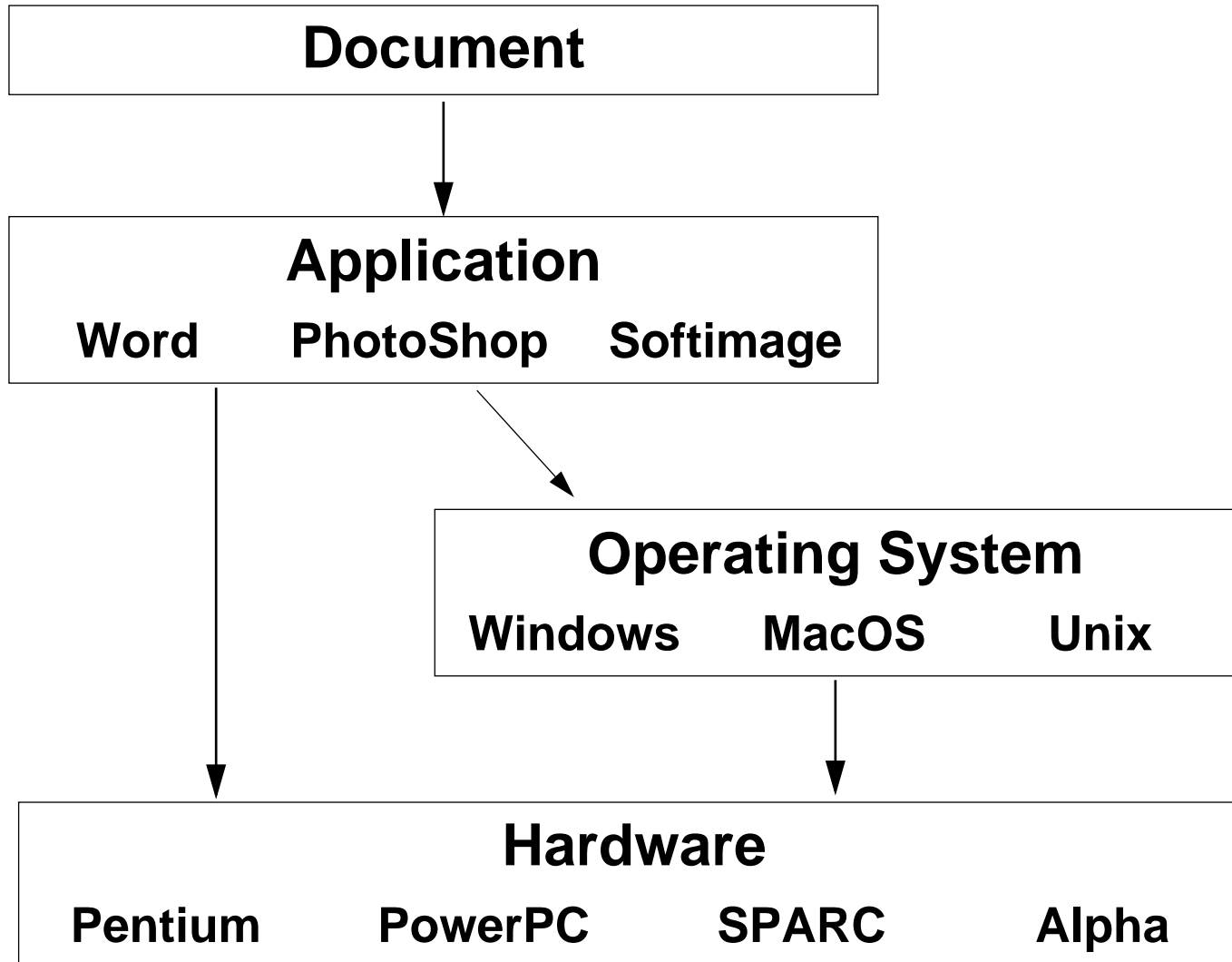
Define:

- **An emulator is something that performs the function of something else.**
 - Emulation is NOT the same as simulation
 - (For example: airplane simulators don't leave the ground)

For our purposes:

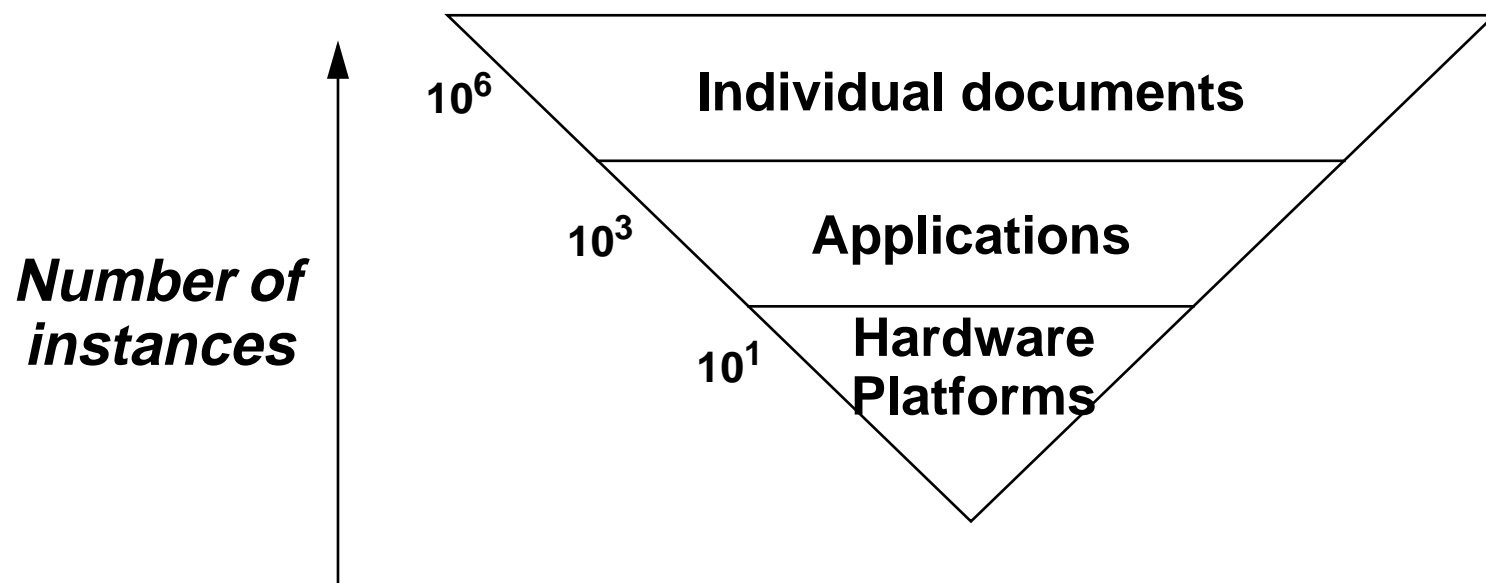
- **An emulator is a program that makes one computer act like another (different) computer**
 - Enabling it to run any program that runs on the other computer
- **So an emulator of an obsolete computer could be run on a future computer**
 - Enabling the future computer to run any program that originally ran on the obsolete computer
- **Note: an emulator is a virtual machine (VM)**
 - i.e., a program (software) that virtually creates a machine (hardware)

Note that emulating an OS buys very little



Why emulate hardware?

- **Hardware is easier to describe (i.e., specify) than software**
 - Since it must be manufactured
- **Hardware is more stable than software**
 - Since it is harder to change or upgrade
- **There is a lot less hardware to emulate than software**
 - Since it is harder to change or upgrade



Hardware specification

REGISTER SETS: 32 general purpose registers

mau := 8; /* byte addressable memory */

regset := R[32] width=32

optype=int,ptr,ptr2,float,double,longdbl,codeptr

regtype=char,short,int,ptr,ptr2,codeptr,long,float,

double,longdbl;

stkptr := R[31];

OPERANDS: describe the operands used in assembly language instructions

operand code_addr codeptr; # pointer to code memory

operand data_addr ptr; # pointer to data memory

operand const16 sconst -32768 32767; # 16 bit signed constant

operand gp_reg reg R; # general purpose register

an "amode" is an address mode. This is a memory reference.

the "ri_addr" amode forms an address by summing a register & a constant.

the "dir_addr" amode forms an address by direct address.

operand ri_addr amode R+const16 format "%B,%O";

operand dir_addr amode data_addr format "%O";

FORMATS: hardware instruction formats use zero or more operands:

"src" means the only source operand; "lsrc" means the left source operand

"rsrc" means the right source operand; "dest" means the destination operand

format mem_load_ri src ri_addr dest gp_reg;

format mem_load_dir src dir_addr dest gp_reg;

format mem_store_rri src gp_reg dest ri_addr;

OPCODES

opcode ldri mem_load_ri;

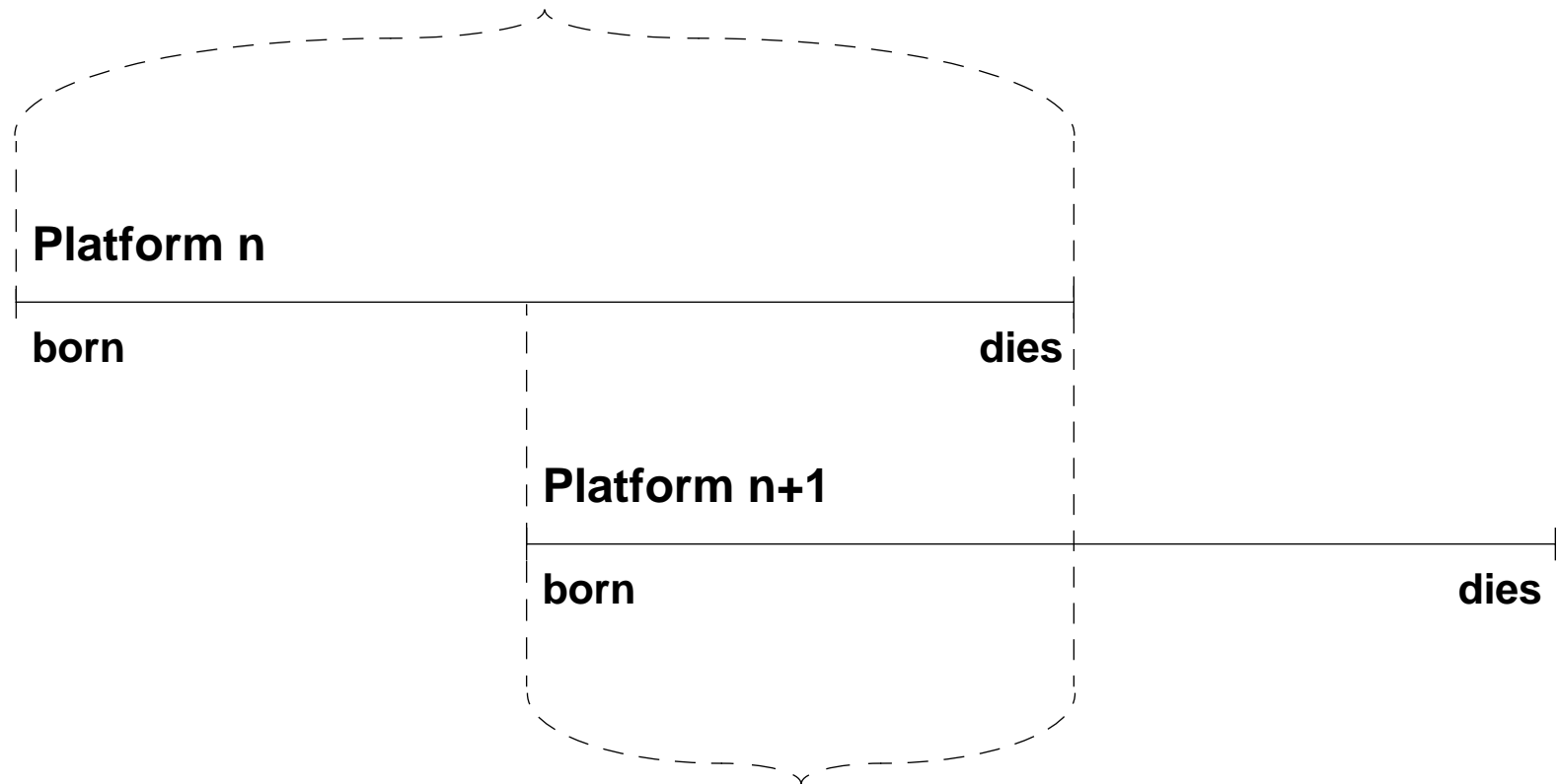
opcode ld mem_load_dir;

opcode jmp : jump branch;

Emulate each platform *before* it becomes obsolete

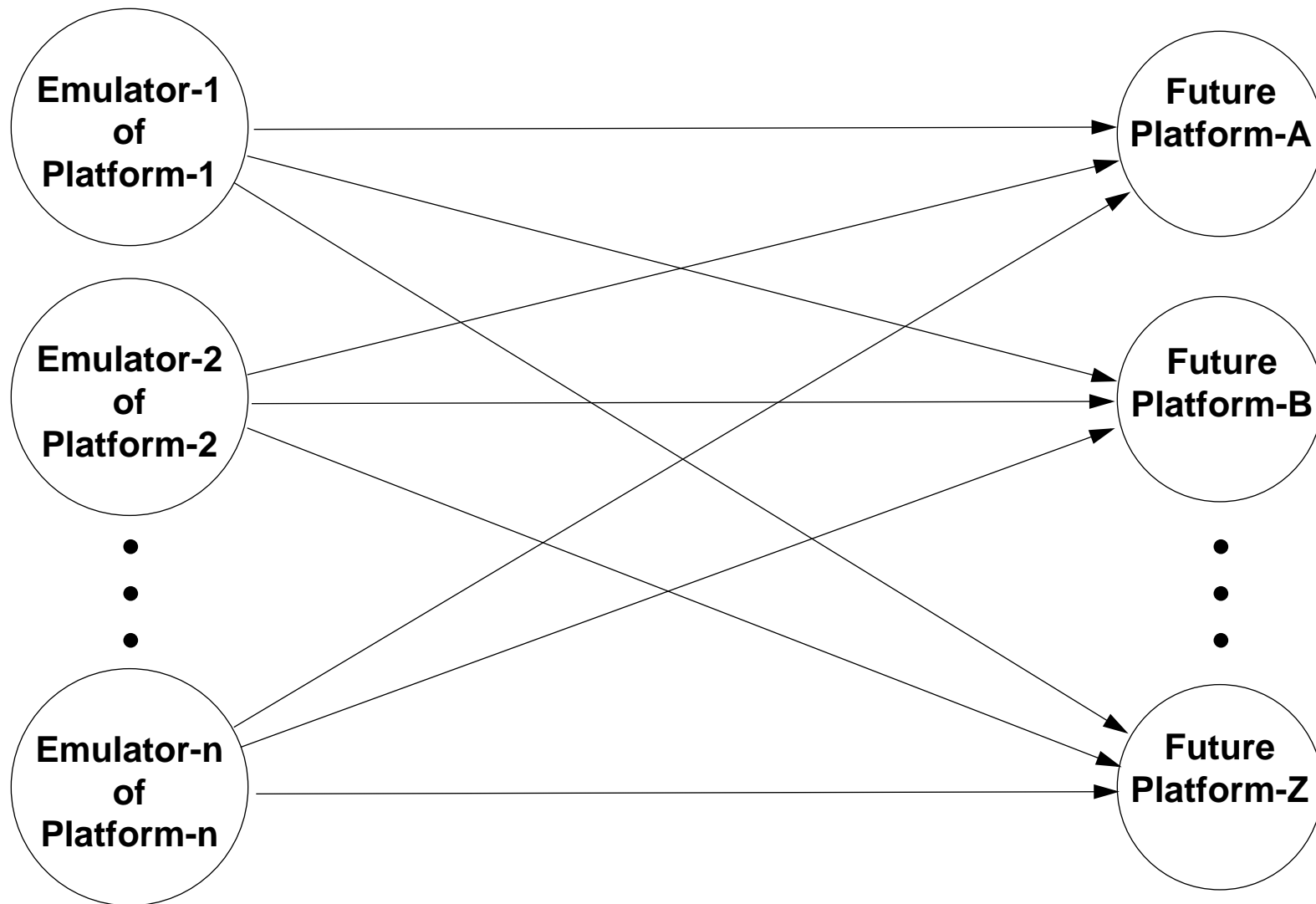


Emulator of Platform n must be written during this time



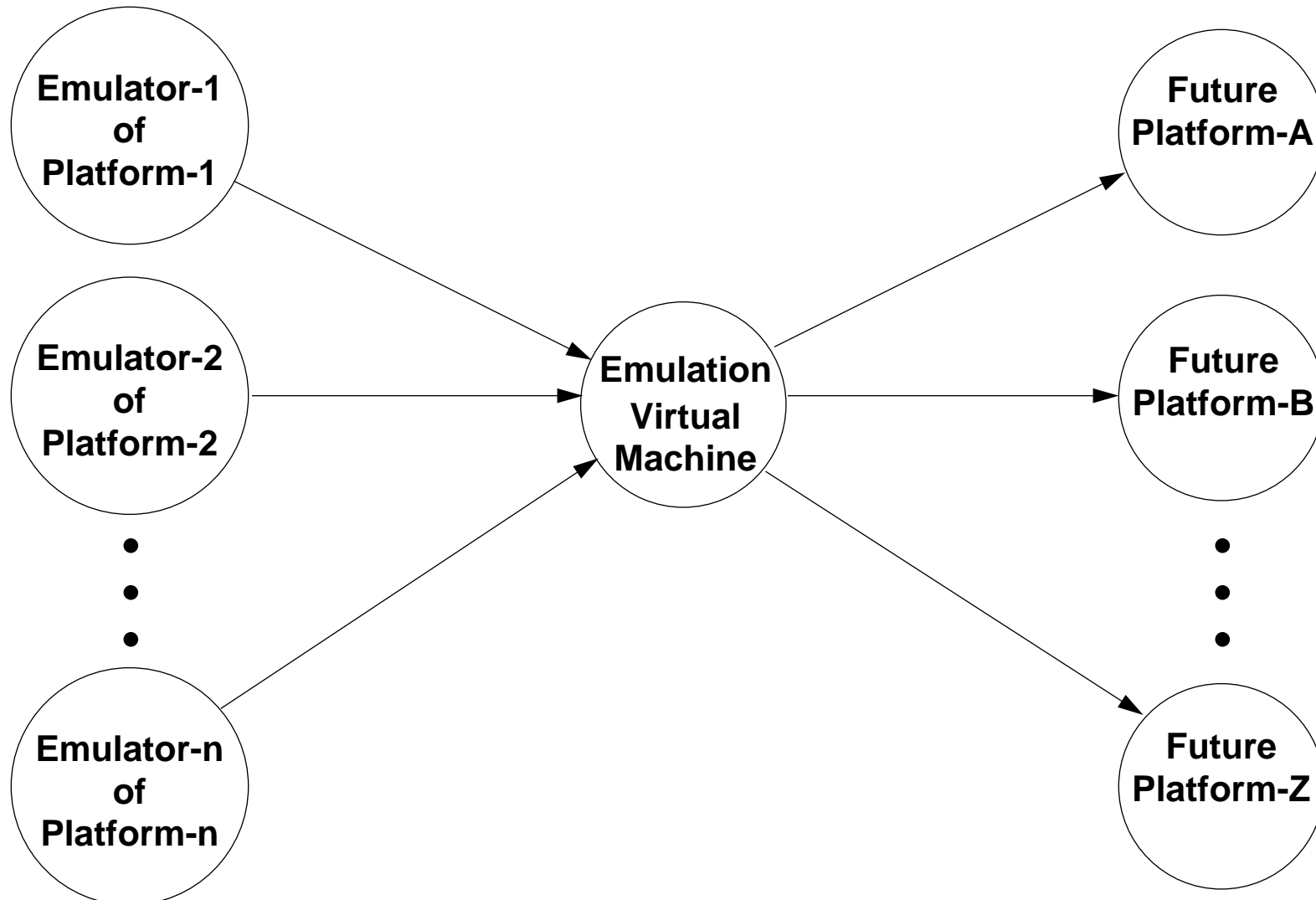
Emulator of Platform n must be ported to Platform n+1 and validated during this time

Avoid having to port emulators to every future platform



Use a Virtual Machine to port emulators to future platforms

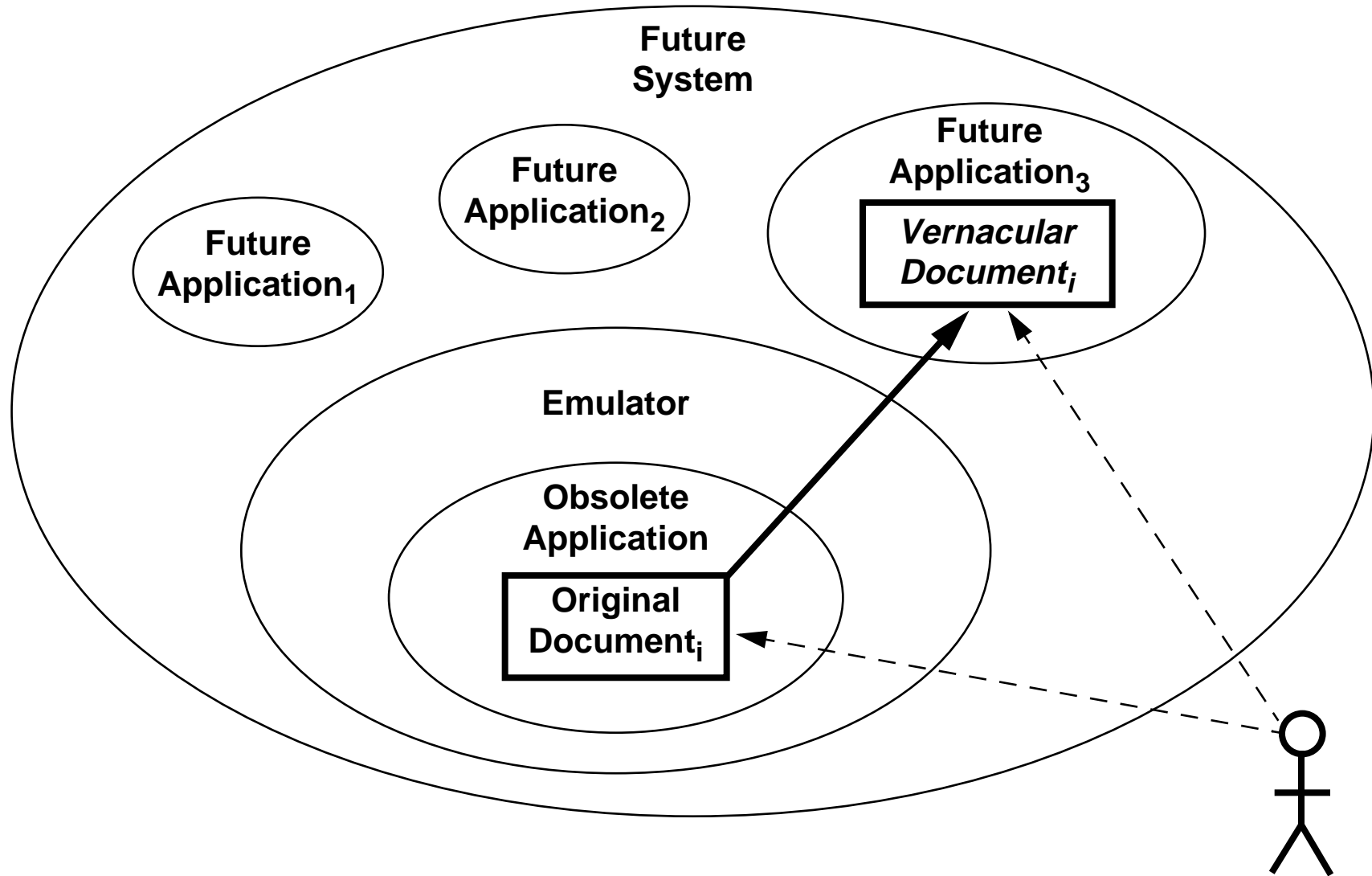
- Write emulators to run on virtual machines, not physical platforms
 - Then need not rewrite all past emulators for each successive platform
 - So long as the virtual machine can be implemented on each platform

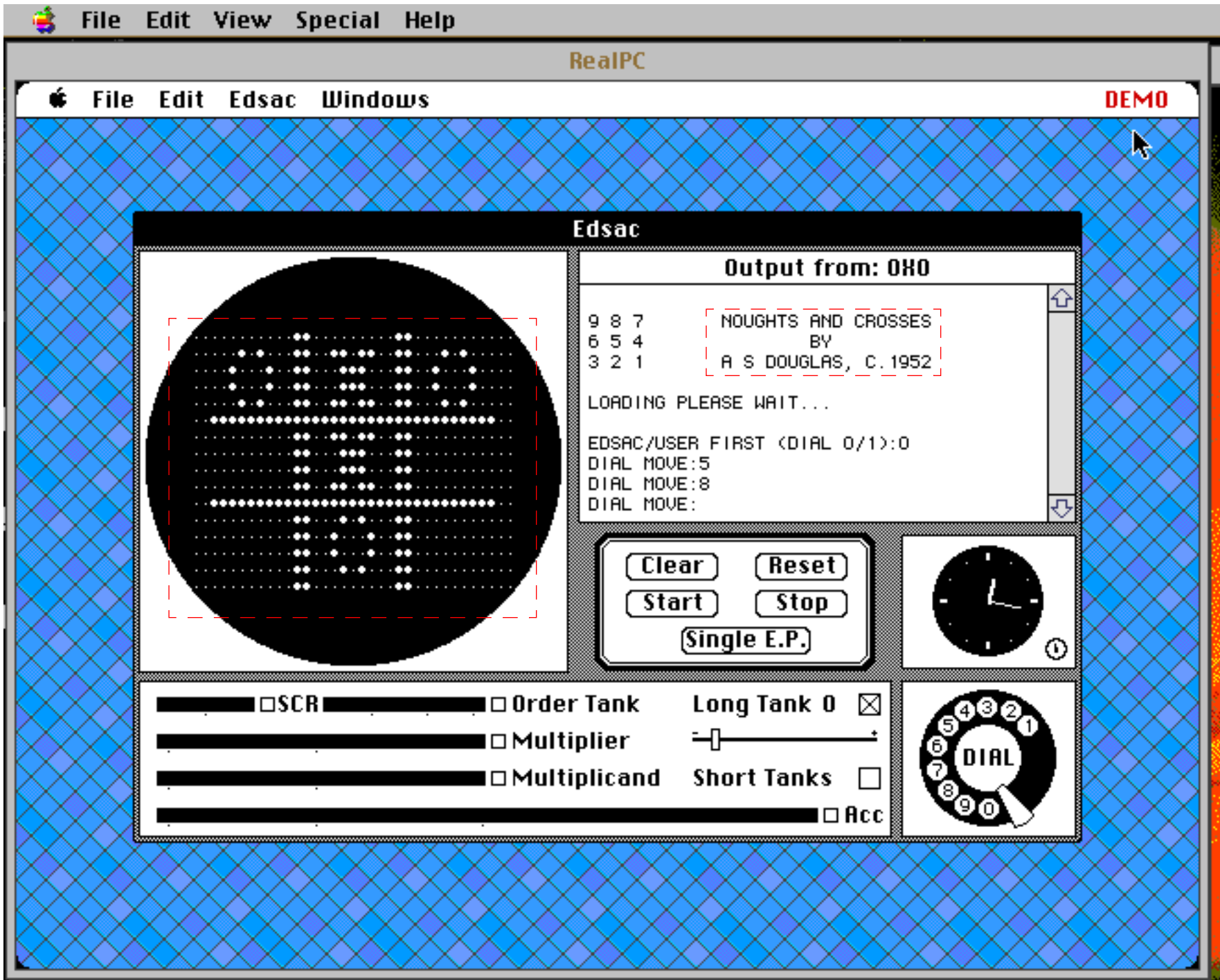


Problems with emulation

- **Users will need to know how to run obsolete software**
 - Need “help” documentation, etc.
 - Analogous to knowing how to read ancient manuscripts
- **May also need “vernacular” use-copies of documents**
 - For non-scholarly access
 - Generate these by conversion, BUT from original, not via intermediate translations
- **Bit streams must not be changed**
 - Corrupted by conversion, compression, inadvertent transformation, etc.
 - Copied imperfectly (i.e., “bit loss”)
- **May have to emulate more than just processors**
 - Graphics/accelerator/sound cards, etc.
 - Displays, peripherals, I/O interfaces
- **Requires one emulator & emulation environment per platform**
 - Either emulate every old machine on every new machine or emulate recursively
 - Get computer vendors or independent suppliers to create emulators

Vernacular extraction from emulation





“Natural Experiments” — Emulation

- **Retro-computing subculture emulates old game systems**
 - Atari, Amiga, Commodore-64, etc.
 - Example: <http://www.jumbo.com/pages/utilities/dos/emulate/>
 - Z80 CP/M emulator for MS-DOS: 52204 bytes
 - Software emulation of 80387 coprocessor chip: 23035 bytes
 - 6800 emulator for DOS, includes a realtime O/S: 56303 bytes
 - Run 8085 Assembly code in your PC (Emulator): 143255 bytes
 - CP/M-86 emulator for MS-DOS: 22748 bytes
 - Apple II emulator for 286+: 191333 bytes
 - Commodore 64 emulator for MS-DOS: 318016 bytes
 - Commodore 64 Emulator for MS-DOS & Win95: 656968 bytes
- **“Backward compatibility”**
 - IBM 360 emulated the older 7090
- **Mac PPC M68000 emulator**
 - Parts of the MacOS itself was emulated through version 8.5 or 8.6
 - M68000 applications continue to run on PPC Macs
- **Transmeta Crusoe processor**
 - Emulates Pentium in microcode
 - Substitutes for Pentium in several systems

Emulation is the only proposed approach that...

- **Can potentially preserve “digital-originals”**
- **Can preserve executable digital artifacts (i.e., “behavioral preservation”)**
- **Can preserve all kinds of digital artifacts in a single, consistent way**
- **Obviates the need to understand the formats of individual documents**
 - **Except what software is needed to view them**
- **Requires zero per-document (artifact) effort, both initially and over time**
 - **Except for copying bitstreams onto new storage media**
- **Defers the need to convert documents into new formats unless and until it is desired to access them in such formats in the future**

But more importantly

- **Migration *cannot preserve originals***
 - It is like “preserving” a painting by copying it—and then copying the copy
 - It provides future access, but not to the original (i.e., not true preservation)
 - It provides access only to future “vernacular renditions” of originals

- **Emulation *does preserve originals***
 - It provides future access to originals—including their behavior
 - It also enables future generation of—and access to—vernacular renditions
 - That is, it provides both preservation and access

Emulation work at the University of Leeds, U.K.

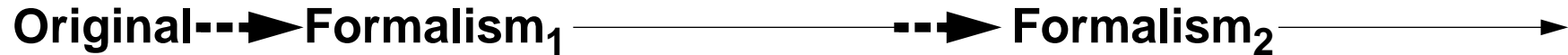
- **The CEDARS & CAMiLEON projects**
 - David Holdsworth, Paul Wheatley, Derek Sergeant, Phil Mellor
 - Working with Margaret Hedstrom, University of Michigan
- **Demonstrating emulation**
 - Of 1980s BBC Domesday system
 - First Algol68 compiler
 - Chukie Egg computer game
- **Holdsworth's emulation philosophy:**
 - Use a HOL as a longevity “platform” (e.g., “C--”)
 - As opposed to a VM
 - Similar to an emulation-specification-interpreter—but compiled, not interpreted
Requires implementing a compiler, not an interpreter on future platforms
- **Recent proposal: “migration-on-request”**
 - Same as Vernacular Extraction concept (introduced above)
 - Retain original by means of emulation, extracting use-copies from it on request

Ray Lorie at IBM Almaden

- **Universal Virtual Computer (UVC)**
 - A first instance of an Emulation Virtual Machine
 - Initial version implemented—but no real I/O
 - Use the UVC to explore two different preservation approaches
- **“Data Archiving”**
 - Write UVC programs to extract the ‘core content’ of digital documents
 - This has been demonstrated on simple data files & PDF files
 - Requires reverse-engineering each format
 - This is something of a cross between formalization & using viewers
- **“Behavior Archiving”**
 - Emulation to reproduce the original behavior of documents (i.e., “render” them)
 - Not demonstrated yet
 - Input/Output not dealt with yet

Process models of preservation approaches

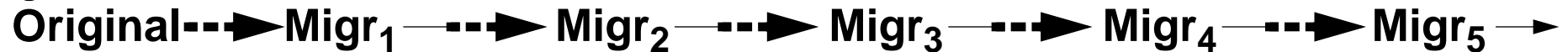
Formalization:



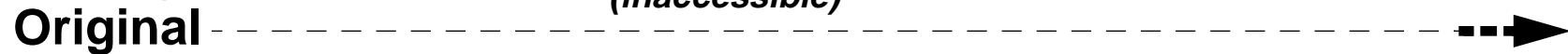
Standardization:



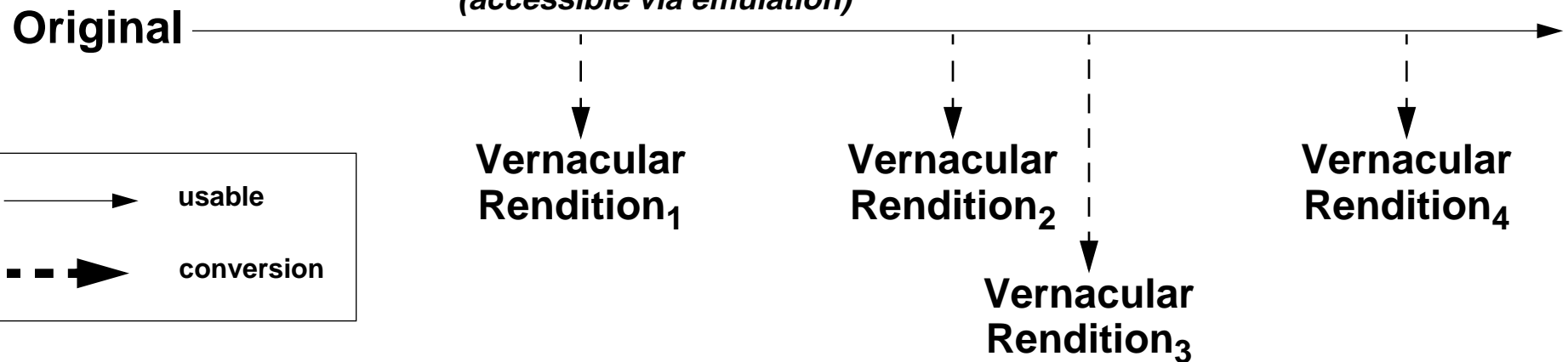
Migration:



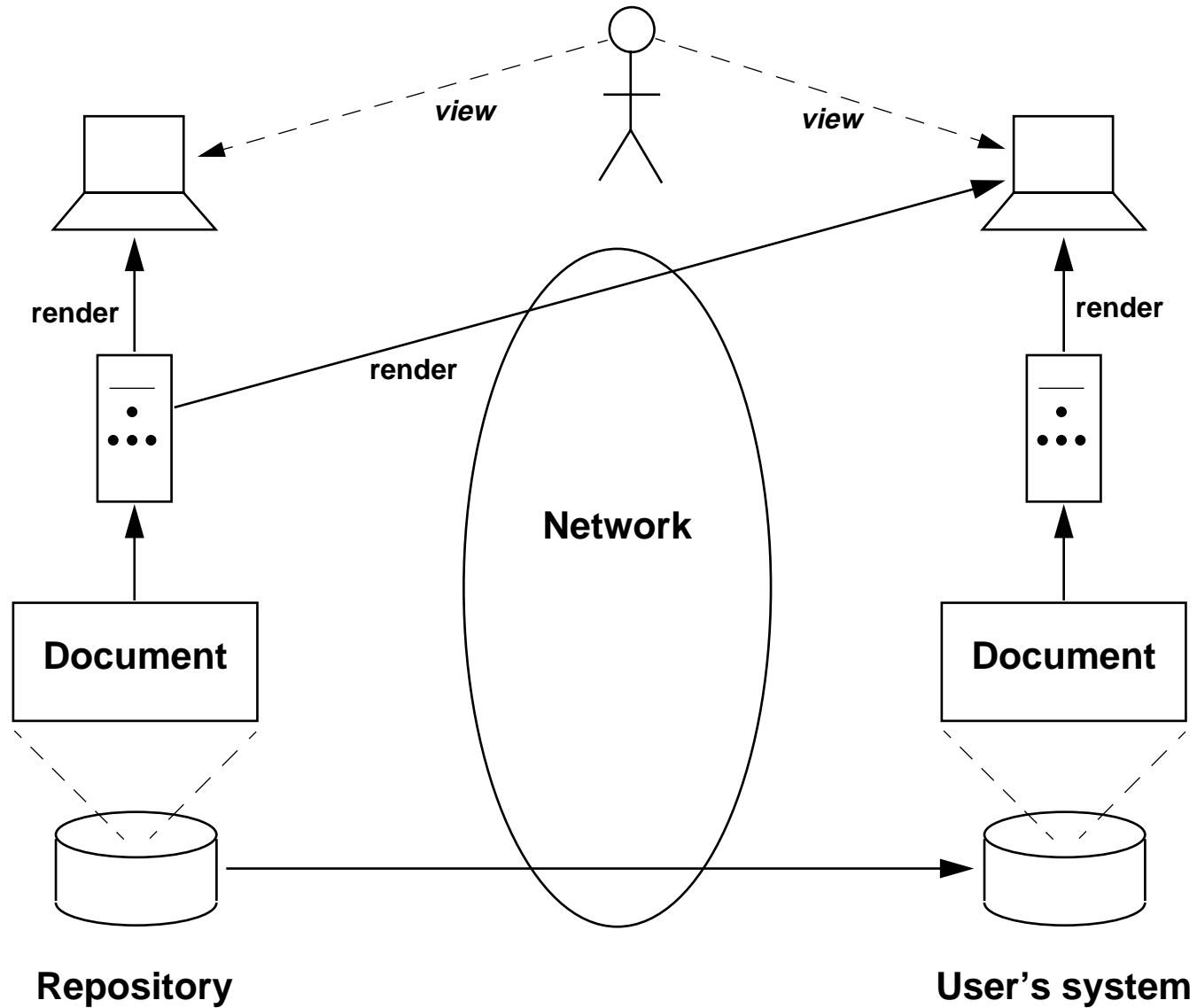
Archaeology:



Emulation:



Alternatives for delivering preserved documents



Expected cost & effectiveness comparisons

| |
|---|
| H,M,L: High, Med, Low +,- : Frequent, Rare |
|---|

| | archaeology | formalizio | standards | viewers | migration | emulation |
|--|-------------|------------|-----------|---------|-----------|-----------|
|--|-------------|------------|-----------|---------|-----------|-----------|

Cost:

Per-approach (x 1)

| | | | | | | |
|-------------------------|---|-----|---|---|---|-----|
| Create EVM or formalism | 0 | H/- | 0 | 0 | 0 | H/- |
|-------------------------|---|-----|---|---|---|-----|

Per-platform (x 10)

| | | | | | | |
|----------------------|---|---|---|---|---|-----|
| Create H/W emulators | 0 | 0 | 0 | 0 | 0 | H/- |
|----------------------|---|---|---|---|---|-----|

| | | | | | | |
|-----------------------|---|-----|-----|-----|-----|-----|
| Port to new platforms | 0 | L/- | M/- | H/- | M/- | M/- |
|-----------------------|---|-----|-----|-----|-----|-----|

Per-format (x 1000)

| | | | | | | |
|------------------|---|-----|-----|-----|-----|---|
| Reverse-engineer | 0 | H/- | H/- | H/+ | H/+ | 0 |
|------------------|---|-----|-----|-----|-----|---|

| | | | | | | |
|----------------------|---|---|---|-----|-----|-----|
| Obtain necessary S/W | 0 | 0 | 0 | M/+ | M/- | L/+ |
|----------------------|---|---|---|-----|-----|-----|

Per-document (x 100,000,000)

| | | | | | | |
|-------------------|---|---|---|---|---|---|
| Process at Ingest | 0 | H | H | 0 | 0 | 0 |
|-------------------|---|---|---|---|---|---|

| | | | | | | |
|-------------------|---|-----|-----|-----|-----|---|
| Convert over time | 0 | M/- | H/- | H/+ | H/+ | 0 |
|-------------------|---|-----|-----|-----|-----|---|

| | | | | | | |
|--------|---|---|---|---|---|---|
| Access | H | M | L | L | L | L |
|--------|---|---|---|---|---|---|

Effectiveness:

| | | | | | | |
|------------------|---|---|---|---|---|---|
| On each document | L | M | M | M | M | H |
|------------------|---|---|---|---|---|---|

| | | | | | | |
|----------------------|---|---|---|---|---|---|
| % of formats handled | L | L | L | M | L | H |
|----------------------|---|---|---|---|---|---|

Open Archival Information System (OAIS)

- **CCSDS: Consultative Committee for Space Data Systems (NASA)**
 - ISO 14721:2002
- **Provides a useful *Reference Model***
 - Defines common terminology:
 - AIP (Archival Information Package)
 - SIP (Submission Information Package)
 - DIP (Dissemination Information Package)
 - PDI (Preservation Description Information)
 - Representation Information / Representation Networks
 - Defines a common framework for talking about digital repositories
 - Discusses many important issues:
 - Ingest formats and processing
 - Use of standards
 - Metadata
- **Though (ironically) says relatively little about preservation**
 - Assumes migration will be used
 - Dismisses emulation as unproven
 - Discusses preserving “look-and-feel” but
 - “Preservation Planning” was added as an afterthought—not well integrated
 - Representation Networks *may* be terminated by Access Software—but disparaged

OAIS Preservation Planning: Assumes Migration

But recognizes that: “Digital Migrations are time consuming, costly, and expose the OAIS to greatly increased probabilities of information loss. Therefore, an OAIS has a strong incentive to consider Digital Migration issues and approaches.”

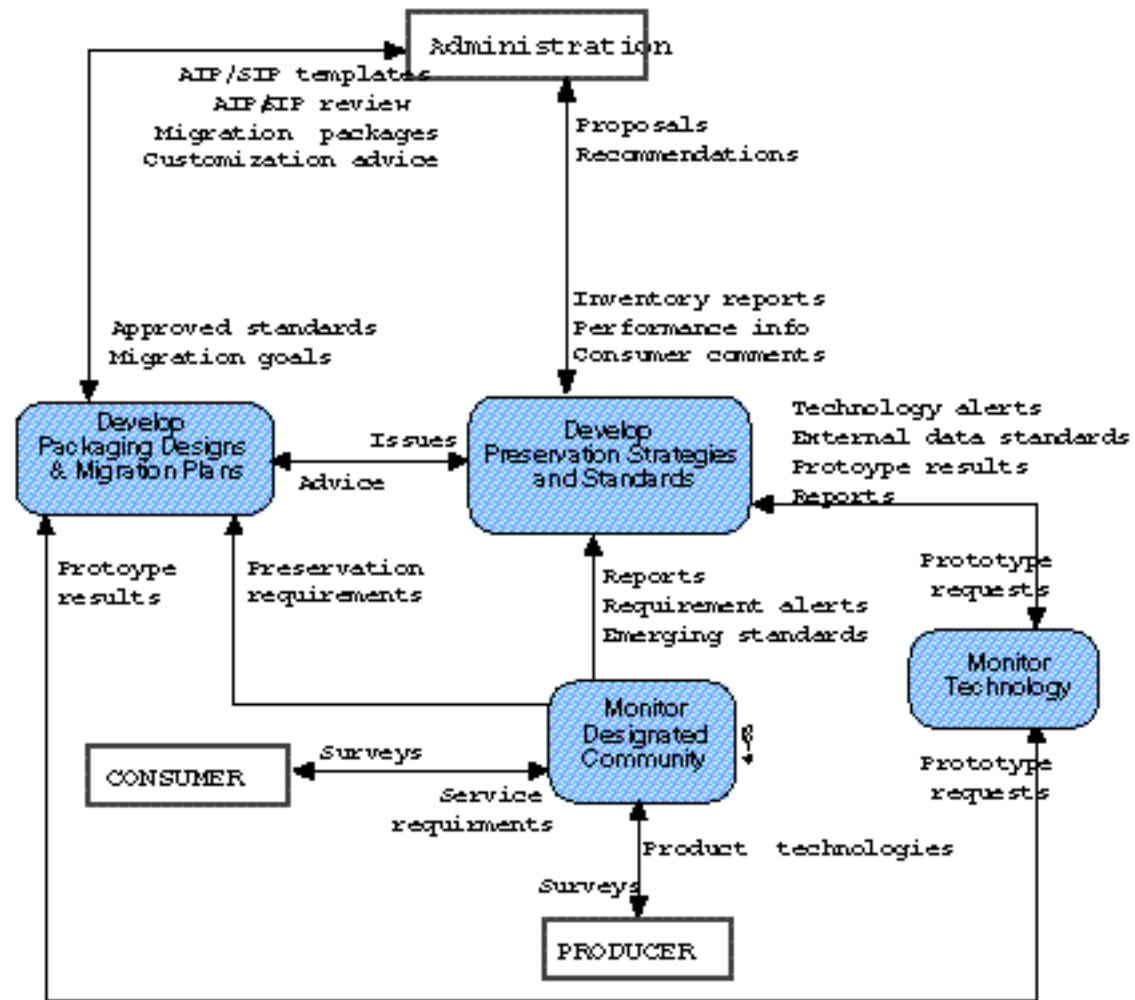


Figure 4-6: Functions of Preservation Planning

Mixed preservation strategies

- **Traditional conservation is both medium-specific & discipline-specific**
 - Treat books differently from paintings, sculpture, textiles, furniture, audio tape, etc.
 - Libraries, archives, museums, scientific repositories, etc. have different agendas
- **But the homogeneity of all digital artifacts creates new possibilities**
 - All digital artifacts can be treated by any of the approaches we have discussed
 - *And* these approaches are not mutually exclusive
- **Using one approach for everything would be simpler**
 - May be unwarranted or too expensive for some kinds of documents
 - *But* it would take advantage of economy of scale, so might ultimately be cheaper
 - Using one cheap approach would be better than using many expensive ones!

A suggested, robust preservation strategy

- **Analyze your preservation needs**
 - Decide which documents don't warrant preservation, which need behavioral preservation
- **Keep the bits alive**
 - Establish policies and procedures to copy bitstreams to new media as necessary
- **Take advantage of formalization & *open* standards where feasible**
 - Without forcing revision of documents or compromising required access or usability
- **Save page-images where sufficient**
 - Using standard formats, if possible
- **Utilize migration as a stopgap measure**
 - Select & pressure vendors to provide migration paths
 - Capitalize on migration when it must be done anyway, i.e., to keep documents “active”
 - Bite the bullet in other cases, when necessary
- **Hedge your bets by paving the way for emulation**
 - Save application & system software (all relevant versions & configurations)
 - For behavioral preservation of originals; or as a low-cost backup, to preserve everything
 - Assuming that suitable emulators will become available