Storage Systems

NPTEL Course

Jan 2012

(Lecture 15)

K. Gopinath

Indian Institute of Science

Indus Script

- Yet un-deciphered: meaning across time not yet accomplished
- Compare with hieroglyphics (Egyptian Rosetta stone): three scripts side by side

- What is the problem? Not enough contextual info:
  - can see the script (human "readable")
  - no mapping between symbols and phonemes
  - need interpretation of sequences of symbols
  - a problem in archaeology, history, society,...

Vedas

Transmitted across atleast 3500-5000 years without differing versions

Including exact pronunciation!

“UNESCO proclaimed the tradition of Vedic chant a Masterpiece of the Oral and Intangible Heritage of Humanity on November 7, 2003”

What "technology" used? Redundancy!

Various "pathas" of Samhita text: can recover from a corrupted text due to added redundancy: RAID-like! (Redundant Array of Indep Disks)

Pada-patha: each word in its separate form

Krama-patha: connects a word in pairs

ABCD becomes AB BC CD DE... (“2-mirroring”): 2 copies

Jata-patha: ABBAAB (“3-mirroring”): 3 copies of A, B, ...

a Ghana-patha (ABBA ABCCBA ABC BCCB BCDDCB BCD...)(“10x”)

Metrical (similar to checksums!) & Musical

"Information dispersal"

Human Reproduction! (Oral transmission)

Use efficient “virtualizers”!

पदपाठ

1,2,3

तम् । भागधेयेन । वि । मुञ्चति । प्रतिष्ठित्यै । यया । रज्ज्वा । उत्तमां । गाम् । आजेत् । ताम् । भ्रातृव्याय । प्र । हिणुयात् । निर्ऋतिम् । एव । अस्मै । प्र । हिणोति ॥ तै सं २-२-६-५ ॥

घनपाठ

1,2,2,1,1,2,3,3,2,1,1,2,3.

तं भागधेयेन भागधेयेन तं तं भागधेयेन वि वि भागधेयेन तं तं भागधेयेन वि ॥

भागधेयेन वि वि भागधेयेन भागधेयेन वि मुञ्चति मुञ्चति वि भागधेयेन भागधेयेन वि मुञ्चति । भागधेयेनेति भागऽधेयेन ॥

वि मुञ्चति मुञ्चति वि वि मुञ्चति प्रतिष्ठित्यै प्रतिष्ठित्यै मुञ्चति वि वि मुञ्चति प्रतिष्ठित्यै ॥

मुञ्चति प्रतिष्ठित्यै प्रतिष्ठित्यै मुञ्चति मुञ्चति प्रतिष्ठित्यै यया यया प्रतिष्ठित्यै मुञ्चति मुञ्चति प्रतिष्ठित्यै यया ॥

प्रतिष्ठित्यै यया यया प्रतिष्ठित्यै प्रतिष्ठित्यै यया रज्ज्वा रज्ज्वा यया प्रतिष्ठित्यै प्रतिष्ठित्यै यया रज्ज्वा । प्रतिष्ठित्या इति प्रतिऽस्तित्यै ॥

यया रज्ज्वा रज्ज्वा यया यया रज्ज्वोत्तमामुत्तमꣳ रज्ज्वा यया यया रज्ज्वोत्तमाम् ॥

रज्ज्वोत्तमामुत्तमꣳ रज्ज्वा रज्ज्वोत्तमां गां गामुत्तमꣳ रज्ज्वा रज्ज्वोत्तमां गाम् ॥

उत्तमां गां गामुत्तमामुत्तमां गामाजेदाजेद्गामुत्तमामुत्तमां गामाजेत् । उत्तमामित्युत्ऽतमाम् ॥

गामाजेदाजेद्गां गामाजेत्तां तामाजेद्गां गामाजेत्ताम् ॥

आजेत्तां तामाजेदाजेत्तां भ्रातृव्याय भ्रातृव्याय तामाजेदाजेत्तां भ्रातृव्याय । आजेदित्याऽअजेत् ॥

तां भ्रातृव्याय भ्रातृव्याय तां तां भ्रातृव्याय प्र प्र भ्रातृव्याय तां तां भ्रातृव्याय प्र ॥

भ्रातृव्याय प्र प्र भ्रातृव्याय भ्रातृव्याय प्र हिणुयाद् हिणुयात्प्र भ्रातृव्याय भ्रातृव्याय प्र

What is needed?

from *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation by Jeff Rothenberg January 1998*

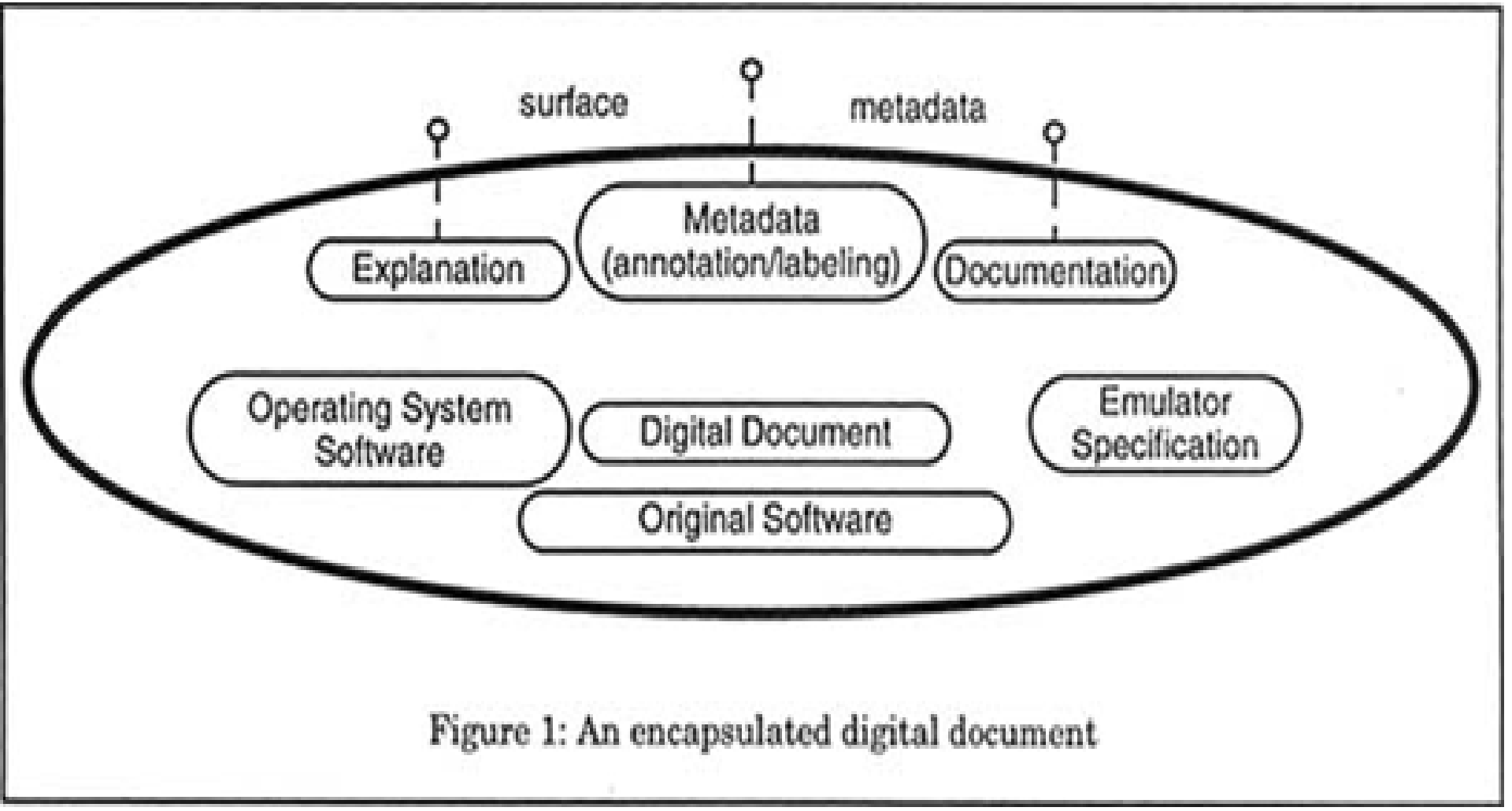

Figure 1: An encapsulated digital document

What is needed? (contd)

from Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation by Jeff Rothenberg

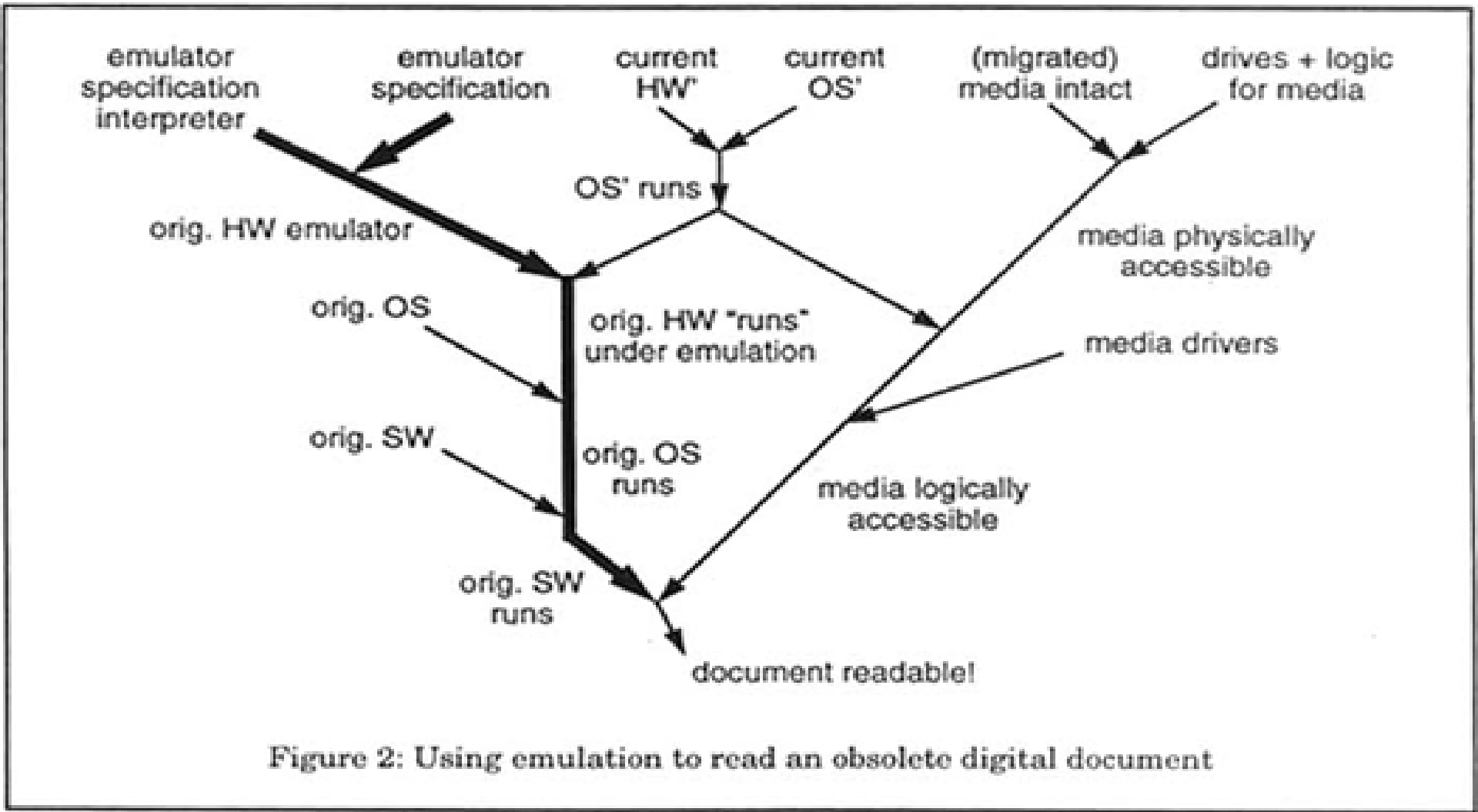

Figure 2: Using emulation to read an obsolete digital document

Film Archival

- Data preservation requires active energy to move it from one format to the next new generation format
  - needs to be done every few years
  - cost is so high that many movies shot digitally stored in analog form as a fallback in case the migration unsuccessful
- Storing a digital master record of a movie costs about $12,514 a year, versus $1,059 to archive a conventional film master in a salt/limestone mine
  - US Academy of Motion Picture Arts & Sciences after a yearlong study of digital archiving in the movie business (2007)

DNA storage (2013)

- Longevity of Data problem:
  - Hardware changes all the time!
  - But not DNA structure!
  - Very compact too
- 2013: Use ternary encoding (A, T, C with G for breaking long seqs of A/T/C as they get missequenced often)
  - For addl redundancy: 100-bases-long DNA inserts staggered by 25 bases so that consecutive fragments have a 75-base overlap
  - storage density about 2.2 Petabytes per gram
    - enough DNA to recover the data about ten additional times

Towards practical, high-capacity, low-maintenance information storage in synthesized DNA (Nick Goldman et al Nature'13)

Insurable Storage

- Digital Documents as insurable property
  - Provide economic incentives for storage service producers and consumers to jointly create a marketplace for a diversity of differentially-priced services
- Insurable Storage Services

Types of Storage

- “Mobile” Storage
  - Memory Stick, Camera, Smart Phone, Laptop
- Personal Storage
  - PC, “Home” RAID systems, “Home” NFS server
- Dept/Organizational Storage
  - NFS/CIFS server
- Cloud Storage
- Highly Available Storage
- Parallel Storage
- Web-scale storage
- Secure Storage
- Attribute-based Storage (“QoS”)
- Long term storage
  - DNA storage!?

Conclusion

- Wide variety of storage designs
  - Each requires different combinations of sw/hw
- Many are still in research stage...

Interfaces to Storage

- Note that Unix tries to treat everything as a file!
  - Hence some “file” ops might not have anything with storage ops
- Libc
  - <stdio.h> in C/Unix but may be arbitrary (for eg, S3 or in parallel user FS)
  - 3 types of buffering:
    - unbuffered, block buffered, and line buffered (default block)
  - FILE structure: a file descriptor, current stream position, eof/error indicators, a pointer to the stream's buffer, if applicable
  - May be able to redefine some functions or load a different shared obj
  - User level fs vs kernel
- System call level: POSIX.1-2008 now
  - Inode (metadata about file)
  - Filesystems may be loadable at runtime (except that contains root image)
- Device Driver
  - Typically loadable at runtime
- Device
- Implications for new models of security/perf...
  - Encryption
  - Mandatory access control/Info flow models
  - Compression

libc

- File access
- *fopen* opens a file
- *freopen* opens a different file with an existing stream
- *fflush* synchronizes an output stream with the actual file
- *fclose* closes a file
- *setvbuf* sets the buffer and its size for a file stream
  - *setbuf* sets the buffer for a file stream
- *fwide* switches a file stream between wide character I/O and narrow

Direct input/output

- *fread* reads from a file
- *fwrite* writes to a file

libc

- Unformatted input/output
- *fgetc getc fgetwc getwc* reads a byte/wchar_t from a file stream
- *fgets fgetws* reads a byte/wchar_t line from a file stream
- *fputc putc fputwc putwc* writes a byte/wchar_t to a file stream
- *fputs fputws* writes a byte/wchar_t string to a file stream
- *getchar getwchar* reads a byte/wchar_t from stdin
- *gets* reads a byte string from stdin (deprecated in C99, obsoleted in C11)
- *putchar putwchar* writes a byte/wchar_t to stdout
- *puts* writes a byte string to stdout
- *ungetc ungetwc* puts a byte/wchar_t back into a file stream

libc

- Formatted input/output
- *scanf fscanf sscanf wscanf fwscanf swscanf* reads formatted byte/wchar_t input from stdin, a file stream or a buffer
- *vscanf vfscanf vsscanf vwscanf vfwscanf vswscanf* reads formatted input byte/wchar_t from stdin, a file stream or a buffer using variable argument list
- *printf fprintf sprintf snprintf wprintf fwprintf swprintf* prints formatted byte/wchar_t output to stdout, a file stream or a buffer
- *vprintf vfprintf vsprintf vsnprintf vwprintf vfwprintf vswprintf* prints formatted byte/wchar_t output to stdout, a file stream, or a buffer using variable argument list
- *perror* writes a description of the current error to stderr

libc

- File positioning
- *ftell* returns the current file position indicator
- *fgetpos* gets the file position indicator
- *fseek* moves file position indicator to a specific location in a file
- *fsetpos* moves file position indicator to a specific location in a file
- *rewind* moves the file position indicator to the beginning in a file

Error handling

- *clearerr* clears errors
- *feof* checks for the end-of-file
- *ferror* checks for a file error

Operations on files

- *remove* erases a file
- *rename* renames a file
- *tmpfile* returns a pointer to a temporary file
- *tmpnam* returns a unique filename

A Common System Interface: Posix 1

- *access* Tests for file accessibility
- *chdir* Changes current working directory
- *chmod* Changes file mode
- *chown* Changes owner and/or group of a file
- *close* Closes a file
- *closedir* Ends directory read operation
- *creat* Creates a new file or rewrites existing one
- *dup* Duplicates an open file descriptor
- *dup2* Duplicates an open file descriptor
- *execl* Executes a file
- *execle* Executes a file
- *execlp* Executes a file
- *execv* Executes a file
- *execve* Executes a file
- *execvp* Executes a file
- *_exit* Terminates a process
- *fcntl* Manipulates an open file descriptor
- *fdopen* Opens a stream on a file descriptor
- *fork* Creates a process
- *fpathconf* Gets config variable for an open file
- *fstat* Gets file status
- *getcwd* Gets current working directory
- *link* Creates a link to a file
- *lseek* Repositions read/write file offset
- *mkdir* Makes a directory
- *mkfifo* Makes a FIFO special file
- *open* Opens a file
- *opendir* Opens a directory
- *pathconf* Gets config variables for a path
- *pipe* Creates an interprocess channel
- *read* Reads from a file
- *readdir* Reads a directory
- *rename* Renames a file
- *rewinddir* Resets the readdir() pointer
- *rmdir* Removes a directory
- *stat* Gets information about a file
- *umask* Sets the file creation mask
- *unlink* Removes a directory entry
- *utime* Sets file access & modification times
- *write* Writes to a file