Automated Characterisation framework

Robert Sharpe, Tessella

Automated Characterisation framework

- Introduction:
  - What to characterise?
  - Why? When?
- How to characterise automatically?
  - File Characterisation:
    - DROID,Jhove etc.
  - Record Characterisation:
    - Role of “components”
  - Role of Technical Registry (PRONOM)

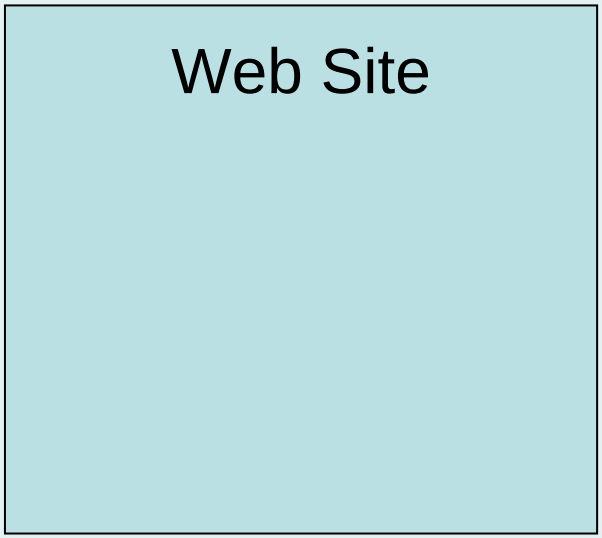
Acknowledgments / Assumptions

- Based on work by Tessella with UK National Archives
  - Part of Seamless Flow programme
  - Automates preservation workflow
- Language based on archives:
  - Applies as well to libraries
  - e.g., in use at British Library
- Deals mainly with migration:
  - Can be applied to emulation
  - May need some changes?

planets

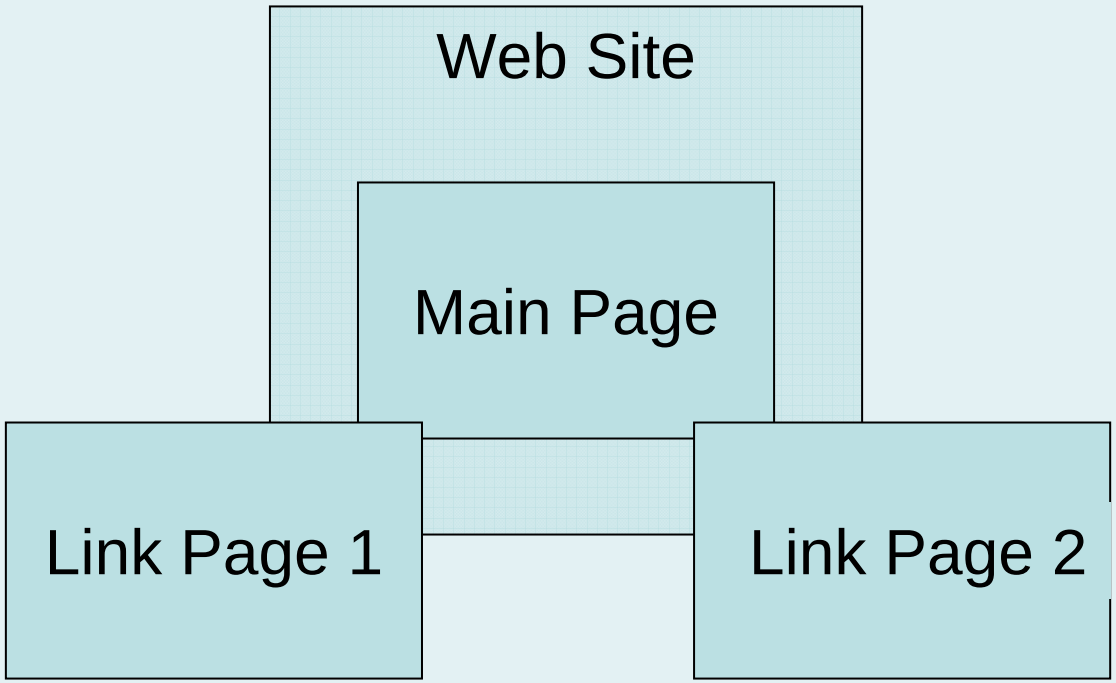
What to characterise?

- Record:



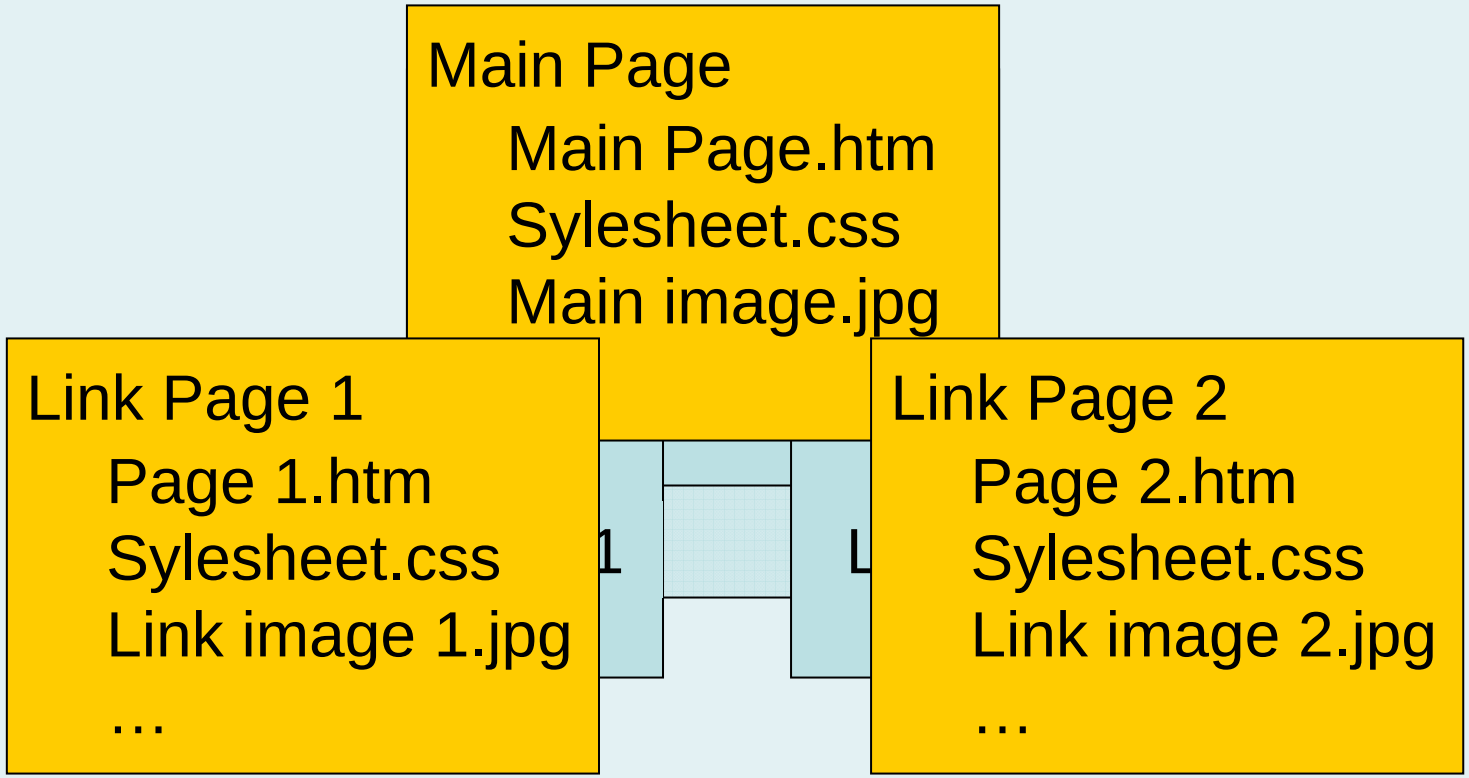
What to characterise?

- Records can be hierarchical:



What to characterise?

- Records can be hierarchical and are made up of files:



Characterising Files: Why? When?

- Why?:
  - Are they in obsolete technology?
    - Depends on format of file
    - Can depend on file properties
    - Will trigger preservation action on record
- When?
  - Whenever a new file enters system:
    - At ingest of a record
    - At migration event

Characterising Records: Why? When?

- Why?
  - Discover “essential characteristics”
  - Measure these characteristics
  - N.B. Will influence file properties to measure
- When?
  - At ingest of a record
  - At migration or emulation event

planets

Characterising Files: Multi-step process

- Fully automated process
- Step 1 – Identification (Find the format):
- Step 2 – Validate format
- Step 3 – Extract properties
- Step 4 – Find embedded objects

planets

Characterising Files: Identification

- DROID (Digital Record Object Identification tool)
- For each format, holds bytestream signature, e.g.,:
  - Header = 474946383961 (i.e. 010001110100100100100 …)
  - Trailer = 3B (i.e. 00111011)
  - Equates to GIF 1989a
- Can be more complicated: missing bytes, variety of options, floating bytestreams etc.
- Information stored in PRONOM
- DROID gets regular updates automatically
- Check every file against every format:
  - List matches
  - Normally just one, sometimes a few
  - Use PUIDs to record identification:
    - GIF 1989a = fmt/4

planets

Characterising Files: Validation

- More detailed check against format specification
- Need a tool per format
- Ask PRONOM which tool to use:
  - e.g. GIF 1989a, use Jhove
- Can update identification, e.g.,:
  - DROID has identical signatures for TIFF3.0 – TIFF6.0
  - But validation tool (Jhove) can tell them apart
  - Run 4 tools, get 1 positive result

planets

Characterising Files: Property extraction

- Again, need a tool per format
- Again, ask PRONOM which tool to use:
  - e.g. GIF 1989a, use Jhove
- Also, ask PRONOM which properties to keep:
- e.g., for GIF 1989a:
  - Compression type — Obsolete check
  - Byte order — Obsolete check
  - Colour space — Obsolete check
  - Image width — Essential characteristic
  - Image height — Essential characteristic
  - Bits per sample — Essential characteristic

planets

Characterising Files: Extract embedded objects

- Again, need a tool per format
- Again, ask PRONOM which tool to use:
  - e.g. ZIP, use unzip
- Run tool
- Characterise these files in turn
- Could lead to iteration

planets

Characterising Records: A problem

- Want to measure essential characteristics:
  - e.g. “Must preserve look and feel”
  - Descriptive
  - Subjective
  - High-level
- Difficult to measure
- Very difficult to measure automatically

planets

Characterising Records: Analyse problem

- Can we break this down?
  - "Must preserve look and feel"
    - Should have 3 Web pages
    - Each page should have 1 image
    - Image 1:
      - Height: 70 pixels
      - Width:104 pixels
      - …
    - Similar for other images…

planets

Characterising Records: Solution

- Break records into “components”
- Record links between components
- Measure properties for each component
- When?
  - At ingest:
    - Identify components and links
    - Measure each component property
  - At migration or emulation event:
    - Check components and links
    - Verify each component property

planets

Characterising records: Solution

- Identify Components
  - Looks through record and decide what is a component, e.g.,:
    - A Web site consists of a series of HTML pages.
    - Each HTML page references images and documents.
  - Build up a “hierarchy” of components with every file associated with a component.
  - Record links between components
- Measure properties:
  - Ask PRONOM for component properties to measure
  - Measure them:
    - If component = 1 file: usually just look up file property
    - If component >1 file, count (e.g., # images)

planets

Role of Technical Registry (PRONOM)

- File Characterisation:
  - Holds format info
  - Holds tools policy: identification, validation, property extraction, embedded object extraction
  - Holds property policy: what to measure
- Record Characterisation:
  - Holds property info and policy
- Preservation planning:
  - Holds format (and property) risks
  - Property tolerance
- Migration:
  - Migration pathways, tools etc.

planets

Summary

- Automated framework:
  - Characterise Files
  - Characterise Records via Components
  - Underpinned by information and policy in Technical Registry
- Part of automated archival process:
  - Provides information for preservation planning
  - Validates migration
  - E.g., Seamless Flow, UK National Archives:
- BUT need:
  - Best practice
  - More tools / better tools / verified tools
  - PLANETS will help…

planets