OCRUI

Interface for the correction
Of OCR text material

OCRUI: *Introduction*

- Who?
  - Juho Vuori† – *frontend*
  - Wouter Van Hemel – *backend*
  - Anis Moubarik – *frontend*

  - Esa-Pekka Keskitalo –

    PAS (Long Term Preservation) team manager
  - Jussi-Pekka Hakkarainen – *project lead*

OCRUI: *Architecture*

Backend <--> Rest API <--> Frontend

The OCRUI application is essentially an editor for the **ALTO XML** format. It consists of a **backend** for managing users, permissions and files, communicating through a **REST API** with a **frontend** interface – the actual editor – for correcting the mistakes often found in OCR'ed text.

→ http://blogs.helsinki.fi/fennougrica/2014/02/21/ocr-text-editor/

OCRUI: *Architecture*

Backend <--> Rest API <--> Frontend

- Python
  - easy to read, hence easy for others to join in
  - lots of programmers to be found
  - specifically uses the minimal Flask web framework
- Postgresql
  - one of the leading SQL databases
  - support for JSON and hstore
    - initially CouchDB considered; SQL + JSON > only JSON
  - advanced text search (tsquery), including dictionaries
- Redis
  - Sessions
  - Statistics (future?)

OCRUI: *Architecture*

Backend <--> **Rest API** <--> Frontend

- Rest: Representational State Transfer
  - GET, POST, PUT, DELETE
- JSON: Javascript data structures
  - fast
  - simple
  - good support in Python and Javascript (duh)

OCRUI: *Architecture*

Backend <--> Rest API <--> **Frontend**

- Javascript
  - codemirror – editing widget
  - backbone.js – light-weight MVCframework
  - others: require.js, underscore.js, bootstrap, jQuery

OCRUI: *In detail – ALTO XML format*

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<alto xmlns="" xmlns:xlink="" xmlns:xsi="" xsi:schemaLocation="">

<Description>
    <MeasurementUnit>pixel</MeasurementUnit>
</Description>

<Styles>
    <ParagraphStyle ID="StyleId-354C3EDE-30F5-45D4-B84B-DF1A56F7D9AB-" ALIGN="Left"
        LEFT="0" RIGHT="0" FIRSTLINE="0"/>
</Styles>

<Layout>
    <Page ID="Page1" PHYSICAL_IMG_NR="1">
    <PrintSpace HEIGHT="2428" WIDTH="1739" VPOS="0" HPOS="0">
    <TextBlock ID="BlockId-4E0658AE-528C-4B7A-8AEC-385915042405-" HEIGHT="1929" WIDTH="1297"
        VPOS="261" HPOS="225" STYLEREFS="StyleId-354C3EDE-30F5-45D4-B84B-DF1A56F7D9AB-">


        <TextLine HEIGHT="53" WIDTH="1282" VPOS="267" HPOS="232">
            <String CONTENT="tramvainijad" HEIGHT="51" WIDTH="269" VPOS="269" HPOS="320"/>
            <SP WIDTH="29" VPOS="269" HPOS="590"/>
            <String CONTENT="linijad." LANGUAGE="veps" CHANGED="false" EIGHT="50"
                WIDTH="144" VPOS="269" HPOS="620"/>

[...]
```

OCRUI: *In detail – database*

- TABLE Users
  - Unsurprisingly, the user accounts of the application
- TABLE Permissions
  - Flags: what can users do in that collection
- TABLE Resources
  - Collections and documents
- TABLE Revisions
  - A ”version” of a document
- TABLE Pages
  - All versions of pages belonging to a document

OCRUI: *In detail – data store*

```
./data/0/4/04f3c3b7d95467242678b20af5a2c3f2
├── images
│   ├── 00001.jpg
│   ├── 00002.jpg
│   └── 00003.jpg
├── pages
│   ├── 1
│   │   ├── f410039708c8683964fca0b0b8d1f68e  (ALTO version 1)
│   │   └── a70c081cc01b3bb8d7c396de1d35ebef  (ALTO version 2)
│   ├── 2
│   │   ├── 04fc350b6f07012122912174831f6323  (ALTO version 1)
│   │   └── b6444fd3b5fb133bb8b2edc70f4c34f6  (ALTO version 2)
│   └── 3
│       └── 095cfb791d330f6be3f914e53650d329
├── source
│   ├── 00001.tiff
│   ├── 00002.tiff
│   └── 00003.tiff
└── thumbs
    ├── 00001.jpg
    ├── 00002.jpg
    └── 00003.jpg
```

OCRUI: *In detail – data store*

- Why not METS for data storage?
  - ➔ *OCRUI backend initially tried to use METS XML*
    - unique as snowflakes
    - performance of large XML file not sufficient or safe enough for a multi-threading multi-user program

OCRUI: *In detail – diff algorithm*

How does the editor handle changes to the text while preserving their metadata?

- The original and current text are represented as sequences of words.
- An *”edit sequence”* is computed that transforms original to current.
  - *Edit sequence* is a series of *add*, *delete*, *replace*, *equal* operations as defined in context of edit distance algorithms.
- The *equal* words acquire bounding boxes of corresponding original words.
- For words that are produced by transforming operations, bounding boxes of corresponding original words are divided among the current ones.
  - Many distinct cases appear depending on which edit operations are used and whether the sequences of edited words encompass or transgress line boundaries.
  - A lot of factors cause small error to division of bounding boxes, but cumulation of error is avoided by always doing the comparison between *original* and *current* words.

OCRUI: *In detail – save*

What happens when the user clicks save?

- Changed pages are uploaded
  - `form/multipart` with only changed pages
- Backend does basic sanity checks
  - any pages, size not null, latest revision, etc.
- Nested context manager wraps file and database code
  - start database and filesystem transaction
  - add revision to Revisions database table
  - write each ALTO file to its MD5 checksum and add checksum to Pages table
  - ✗ on exception, get rid of already written files if they didn't exist before (remember ”revert”) and roll back database transaction
  - ✓ on success, commit transaction
- Profit!

OCRUI: *Future*

- Multi-user UI improvements
  - better handling of editing conflicts
  - improve revision selection (revert)
  - notes for inter-user communication
  - flag words as ”not sure”
- PDF export of corrected text
- Corpus building
- Dictionaries
  - suggestions in ALTO XML attributes
    (useful or merely cognitive overload?)
- Automatic correction of common OCR mistakes
  - risky – some level of manual intervention?
  - per collection, per language, per script?

OCRUI: *Questions?*

Thank you!