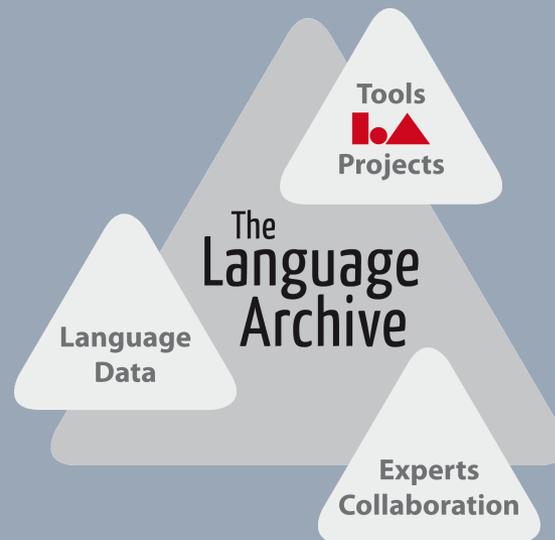# AUVIS Recognizer Dongle, Web Service & ELAN Usability

Eric Auer <eric.auer@mpi.nl> – 1/2015
The Language Archive – Max Planck Institute for Psycholinguistics
Nijmegen, The Netherlands

- Why dongles?
  - Use recognizers in the field, without internet
  - Prevent uncontrolled duplication of recognizer software

- How to use dongles & offline recognizers?
  - Programmer combines recognizer with dongle & toolkit
  - User acquires dongle (unique secure hardware token)
  - User acquires digital license, linked to dongle
  - Programmer distributes dongle-ized recognizer
  - Can use recognizer in ELAN as long as dongle is plugged
  - License based on timespan, usage or concurrency count
  - Copies of the recognizer can only be used with licenses

- Sounds good, so why not use them more often?
  - See next slide…

Recognizers with Dongles - Practice

- How to really use recognizers with dongles...
  - Programmer registers as software manufacturer with dongle provider, acquires key to sign license requests
  - Programmer compiles recognizer with dongle library
    - User needs unique dongle hardware token
    - User installs toolkit from dongle manufacturer
    - Runs toolkit, connects dongle, selects license update
    - Selects license request
    - Selects add software manufacturer
    - Inserts customer number of programmer
    - Emails generated request file to programmer
      - Programmer decides about request, creates signed license
      - Programmer mails license(s) signed for the user's dongle
        - User runs toolkit, connects dongle, updates licenses
        - Selects load license update
        - Adds file received from programmer to dongle
        - Installs recognizer (comes with another copy of the toolkit)
        - Copies CMDI metadata XML file to ELAN extensions dir
        - Edits CMDI file to point to install location of recognizer
        - Runs ELAN to use recognizer
        - Recognizer checks signed license in secure dongle store
        - Runs if in licensed period (dongle and / or online clock)

- How to distribute the efforts of dongle usage
  - Programmer only has to set up workflow once, so average effort over time is small for programmer
  - Helpdesk can pre-install licenses on dongles
  - Programmer decides whether user is given a dongle
  - License expiry for example 1 year after pre-installation
  - Not as good as 1 year after user request, but sufficient

  - Recognizer installer includes dongle toolkit
  - (done) Installer puts recognizer at predictable location:
  - Therefore no CMDI edit by the user necessary
  - If recognizer installer could find ELAN install location…
  - Maybe check common locations or scan registry?

  - Ideally, user just gets pre-licensed dongle, runs installer
  - ELAN runs recognizer while license in the dongle is valid
  - Programmer or helpdesk could remind user about expiry

Recognizers as web service - old

- Why web service?
  - User always gets to invoke newest algorithm
  - Prevent uncontrolled duplication of recognizer software
  - Easy installation: Same invocation for all users

- What are the problems?
  - Algorithm changes may involve user interface changes
  - Have to stay compatible with users with old UI configs
  - Limited ability to provide feedback to the user
  - High demands on network bandwidth for large media
  - High CPU load or RAM on server possible
  - Repeated runs also multiply network I/O load
  - No protection against deliberate waste of host resources

- Sounds interesting, so what could be improved?
  - See next slide…

Recognizers as better web service

- What can be improved for the user?
    - Algorithms can assume defaults for missing parameters
    - Heuristics can be better than user-adjustable settings
    - More real-time recognizer log display now in testing
      (limited gain if recognizer wraps non-AUVIS tools)

- What can be improved for the network?
    - Smart client & caching server could negotiate (e.g. by checksum) about already uploaded media, less I/O use
    - Client could request analysis of files known to archive local to server: Send server Handle or URL, not the file

- What can be improved for the server?
    - Server configured to allow only few concurrent requests
    - Limit CPU time & RAM per request, dedicated temp disk
    - Access to more computational resources could be bound to user login for the web service (also for archived files)

ELAN improvements for recognizers

- More straightforward ways to buttons
  - Hide advanced parameters while in easy mode, unfold or pop-up the extra parameters in advanced GUI mode
  - All recognizers in 1 sorted list, by name & input type
  - Streamline workflow by reducing number of windows / dialog boxes opening, allow more keyboard control

- More awareness of the process
  - More real-time recognizer log display now in testing (both for locally installed & web service recognizers)
  - Improve fault detection & user notification in ELAN

- More control of the process, but keep user effort limited
  - Templates for I/O file placing (tiers...), e.g. 1. same dir as media file, 2. same dir as EAF file, 3. user's temp dir
  - When running the same recognizer several times, make it easy for the user to replace & compare (?) results

ELAN improvements for annotators

- Fast, keyboard-assisted workflow to correct, insert & remove annotations: recognizer results always need edits
- Easy visual comparison of tiers & their relations, also tools to for example move or copy items by annot. text
- Segmentation mode & transcription mode to quickly create & edit rough annotations in optimized workflow

- To quote Lewis Carroll:

    - Would you tell me, please, which way I ought to go from here?
    - That depends a good deal on where you want to get to.

- Thank you for your attention!

- Suggestions & questions are welcome…