Alexandre Arkhipov
Moscow State University

XML technologies in language documentation workflows

PARADISEC: RRR conference
Melbourne, December 4th, 2013

Greetings from Moscow archive

M.V. Lomonosov Moscow State University

Department of Theoretical and Applied Linguistics (OTiPL), Philological faculty

- Founded in 1960
- Fieldwork in minority languages of USSR since 1967 led by **Aleksandr Kibrik**†
- Main destinations: Caucasus (esp. Daghestan), Kamchatka, West Siberia, Volga region, Russian Far East

Greetings from Moscow archive

LangueDOC archive

- In 2005, an NSF-funded project “Five languages of Eurasia” (*PI* A. Nakhimovsky) was launched to create audio-video documentation for selected languages
- In 2008, a dedicated server with LAT software suite was installed at Moscow State University to host the project archive as well as contributions from other research teams
- In 2013, we are hosting data from several Moscow teams, St Petersburg, Tomsk on a dozen of languages including Russian Sign Language

OUTLINE

I XML and related technologies:
advantages and pitfalls

II Data transformations with XSL

III Challenges for a general interlinear format

IV Dynamic annotations

V Outlook

I Data formats

Ubiquitous XML

- MS Word (.docx)
- OpenOffice / LibreOffice (.odt)
- ELAN (.eaf)
- EXMARaLDA
- SpeechAnalyzer (.saxml)
- SIL FLEx (LIFT: lexica)
- SIL FLEx (.flextext: interlinear texts)
- SVG graphics
- KML maps...

I XML: advantages

XML expansion just happens; what are the advantages for us?

- open standard format, viewable & editable in any text editor
- readable by human
- transparent structure
- each user or user group or project or tool can introduce their own format (tag set) to fit their needs

I XML: disadvantages

The weaknesses of XML have mostly the same origin as its strengths:

- verbosity
- high processing time/memory load
- each tool its own format: need for conversion

I XML: remedies

Ways to overcome these weak points:

- verbosity => equivalent more compact XML formats; JSON, YAML
- high processing time => multiple auto-generated representations of the same data for different purposes. Cf. Moran 2012: plain tab-delimited text file; relational tables; RDF/XML
- variety of formats => for each data type (lexicon, text, …) a standard format
+ for each tool, import into own format
(better than pairwise export-import solutions)
(Han Sloetjes p.c.)

I XML: advantages (2)

The entire workflow can go within XML.

Before:

- data entry -> Word
- data analysis (annotation) -> Toolbox
- data storage -> txt
- data publishing -> HTML?
- data retrieval -> plain text search?
- data update -> goto Toolbox
- data reuse -> Word

I XML: advantages (2)

The entire workflow can go within XML.

Or:

- data entry -> Word/Excel
- data analysis (annotation) -> Excel
- data storage -> MySQL
- data publishing -> HTML+PHP
- data retrieval -> +MySQL
- data update -> HTML+PHP
- data reuse -> ?

I XML: advantages (2)

The entire workflow can go within XML.

Easier:

- data entry -> ODT (XML)
- data analysis -> ELAN, FLEx (XML)
- data storage -> just any XML
- data publishing -> XHTML+XSL (XML)
- data retrieval -> XQuery
- data update -> XQuery, XForms
- data reuse -> ODT (XML)

I XML: advantages (3)

XML — RDF — LLOD

- XML formats allow easy transition to RDF (Resource Description Framework), the pillar of the Semantic Web
- RDF allows to apply logical inference adding new data (statements) to the existing ones (database => knowledge base)
- RDF allows to link various sources of information with different internal structure => single search across different sources
- Linked Open Data (LOD), Linguistic LOD

I XML: that simple?

Despite the simple underlying principles, it can appear not so easy to implement complete solutions (e.g. for linguistics) since they may require many different components:

XML, XSLT, XSL-FO, XPath, XQuery, XForms, XML Namespaces, RDF, OWL,...

However, XML technologies are a powerful tool and play well together. Also, as they share the same basis, the learning curve is not so steep.

I XML: databases

Native XML databases

eXist-db, BaseX — free & open-source

- storage
- publication
- search
- update

via rich browser-based applications

both on local and remote computers

II Using XSL Transformations in language documentation

- At the beginning of the «Five languages...» project (2005), we used Toolbox for glossing, BoxReader and MannX (both by Tom Myers) for conversion to HTML and display
- Word documents were used as an medium for collaboration (reviewing & comments)
- MannX, BoxReader and Toolbox were gradually replaced by ELAN and SIL Fieldworks (FLEx)
- ...Which is why we had to use a dozen XSL transforms between various tools and formats

II XSL Transformations (1): Directions

- (BoxReader, in Java):
  Toolbox => HTML (nested <span>s)
- HTML => enhanced HTML
- HTML => OpenOffice ODT
- ODT => HTML

II XSL Transformations (1): Operations

- rearrange tiers
- insert additional tiers (from plain text files)
  e.g. additional translations, narrow phonetic transcription, cyrillic orthography
- insert time offsets (from simple xml files)
- move infixes to their original position in the word
  e.g. "barxar" ‘(donkey) lies down’ {b-**axa**-r-r} => {b-**a**<r>**xa**-r}
- change caps in glosses to small caps (HTML => ODT)
- merge multiple tables into one (for long sentences) (ODT => HTML)
- hide or display comments

II XSL Transformations (2): Directions

- FLEx XML (flextext) => ELAN EAF
- FLEx-exported ODT (with frames) => extract certain tiers into plain text or csv
- FLEx-exported XML (flextext) => extract individual texts from a single flextext file
- ODT => flextext (for "old" texts edited in Word; to make EAF; FLEx did not yet have interlinear import)
- ODT => flextext (for texts prepared for paper publication; to make EAF)
- ODT => flextext (for "old" texts edited in Word; to actually import into FLEx; does not yet use word and morpheme)
- ODT => flextext (for texts transcribed and translated by Archi consultant; to actually import into FLEx)

II XSL Transformations (2): Operations

- tokenize words into morphemes based on morpheme breaks
- (flextext => EAF): cleanup punctuation:
  omit punctuation "word" elements; create phrase-level text items containing words and punctuation concatenated
- (flextext => EAF): handling multiple notes:
  replace 'lang' attribute with consecutive number => each note goes to separate tier

(ODT => flextext): correct styles:

- if more than one translation line per sentence,
  put all but the first as notes (otherwise discarded by FLEx)
- strip all internal formatting (text:span's)
- trace automatically created styles to original style names
- (ODT => flextext): extract info: time offsets, speakers, comments on turn-taking; (re)number sentences

II XSL Transformations (3): To-Do

- Make latest XSLs customizable (pass Office style names and target tier/writing system attributes as parameters)
- Make them available online via eXist-db and web forms

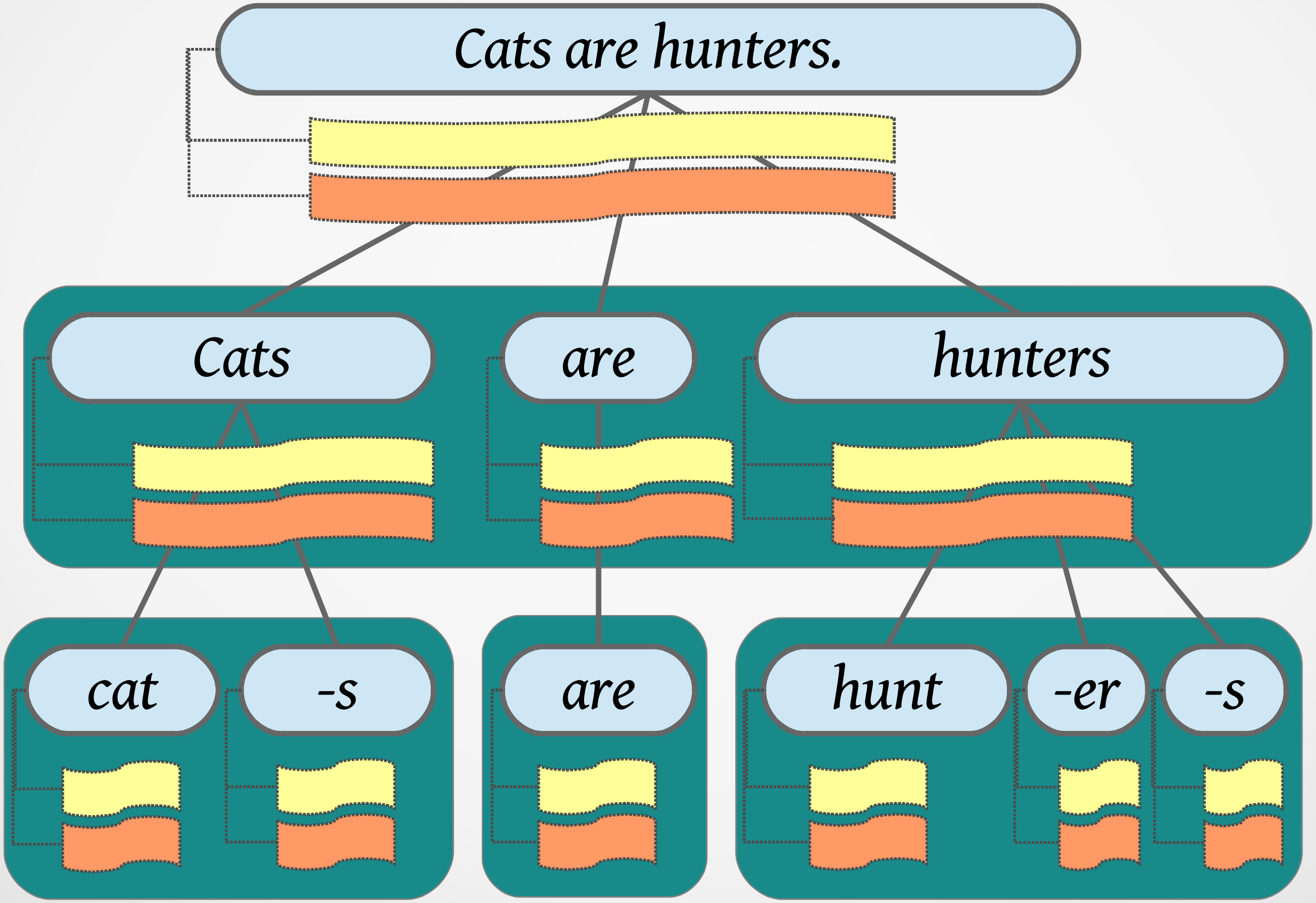
III Challenges for a general interlinear format

- Recall: a standard format for each data type (lexicon, text, ...) + for each tool, import into own format & export into standard format
- Each tool can be specialized and only work on a subset of the general format (e.g. ELAN -> media annotation, FLEx -> glosses)
- Each tool should be able to update its part of the data integrating it into the bigger common data (without breaking anything...)

III Challenges (1): multiple axes

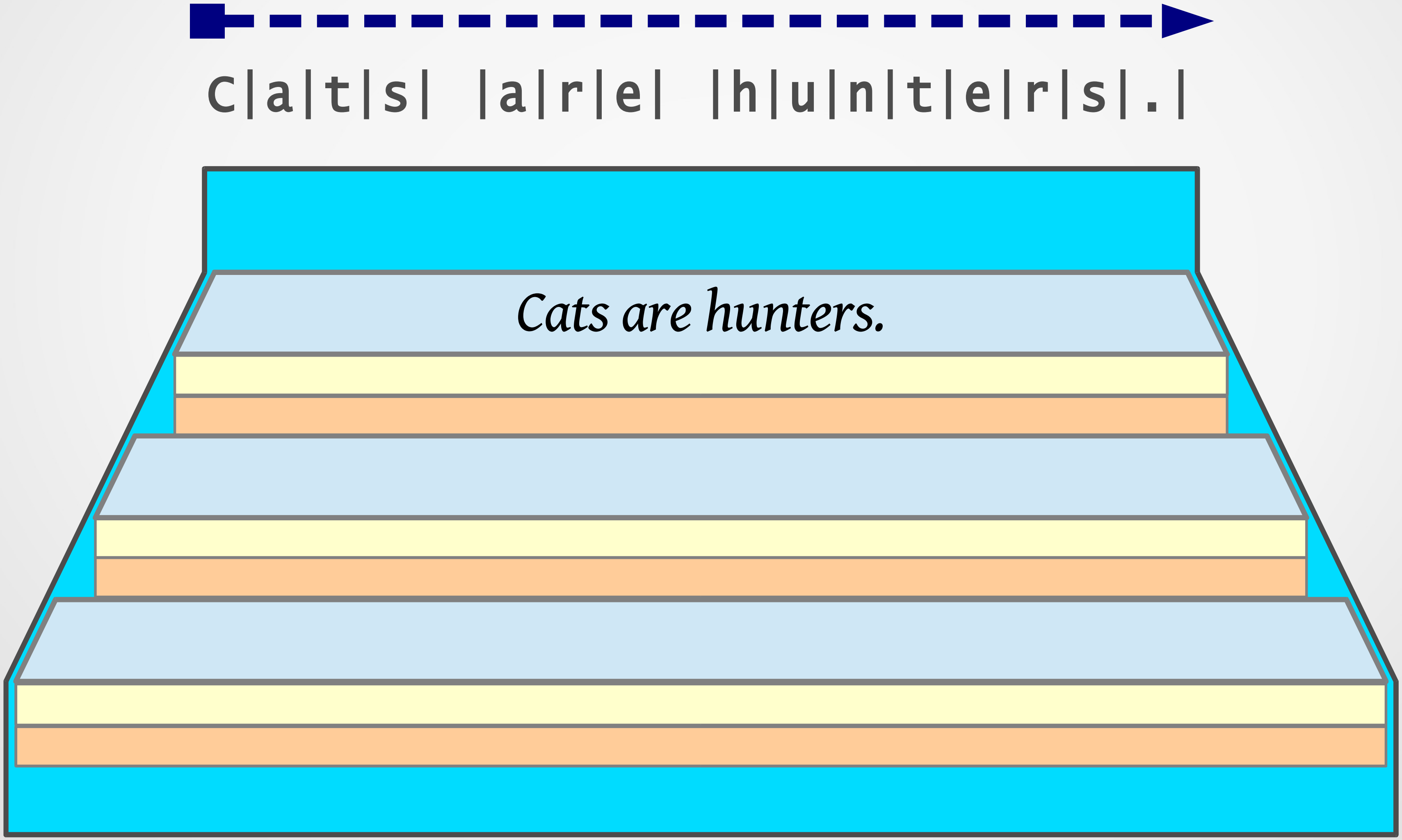
- Multiple axes
  (basic representations), e.g.
  - audio + video + transcript + grammatically normalized text
  - published document/manuscript + grammatically normalized text
  - sign lg + spoken lg
  - BOLD-style: audio annotations
- Each axis can have an analysis tree associated with it
- Each axis can have its own units
- Axes can be aligned to each other

III Axis types & units: Abstract text



III Axis types & units: Abstract text

Plain text axis (character, line)

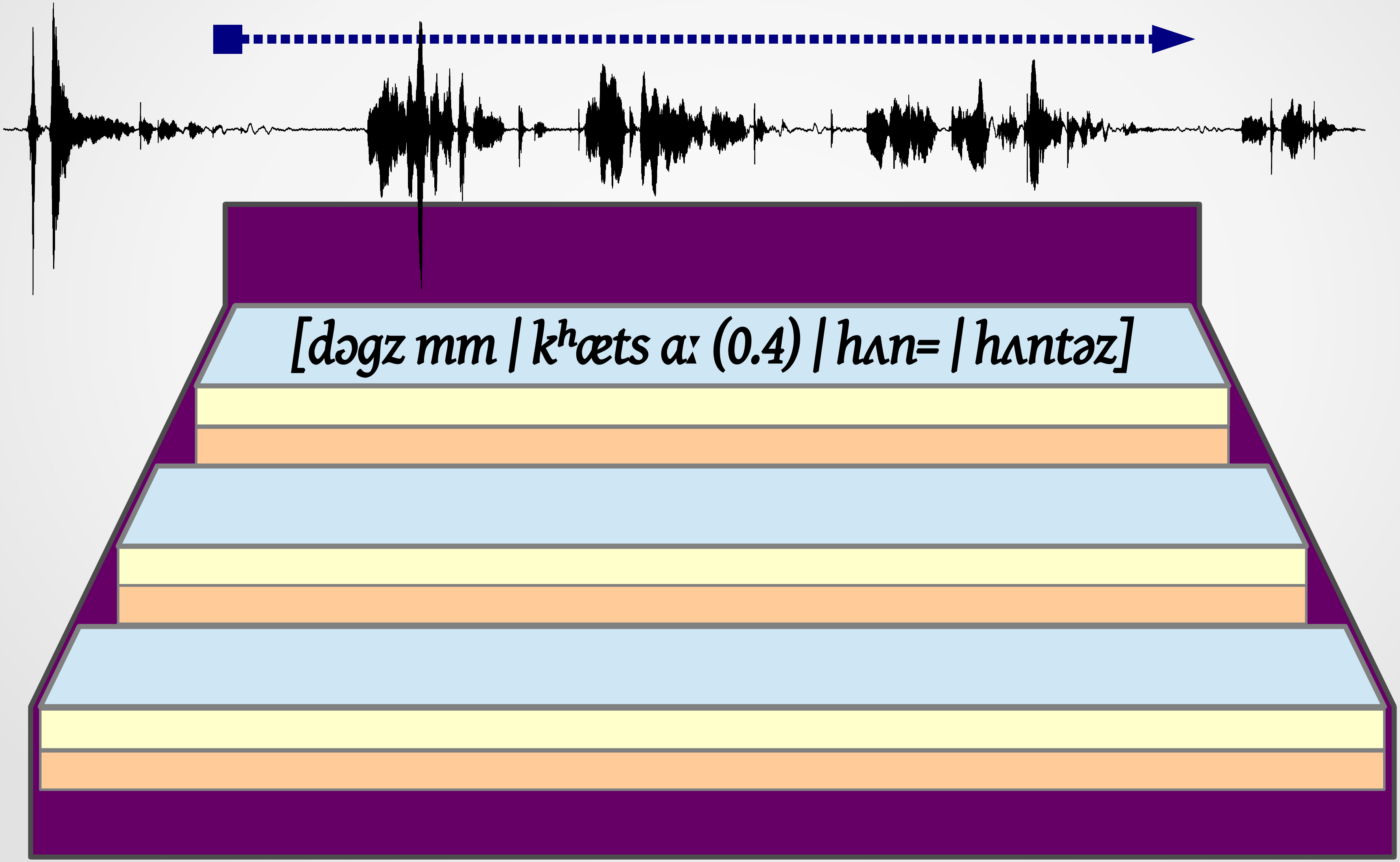


C|a|t|s| |a|r|e| |h|u|n|t|e|r|s|.|

Cats are hunters.

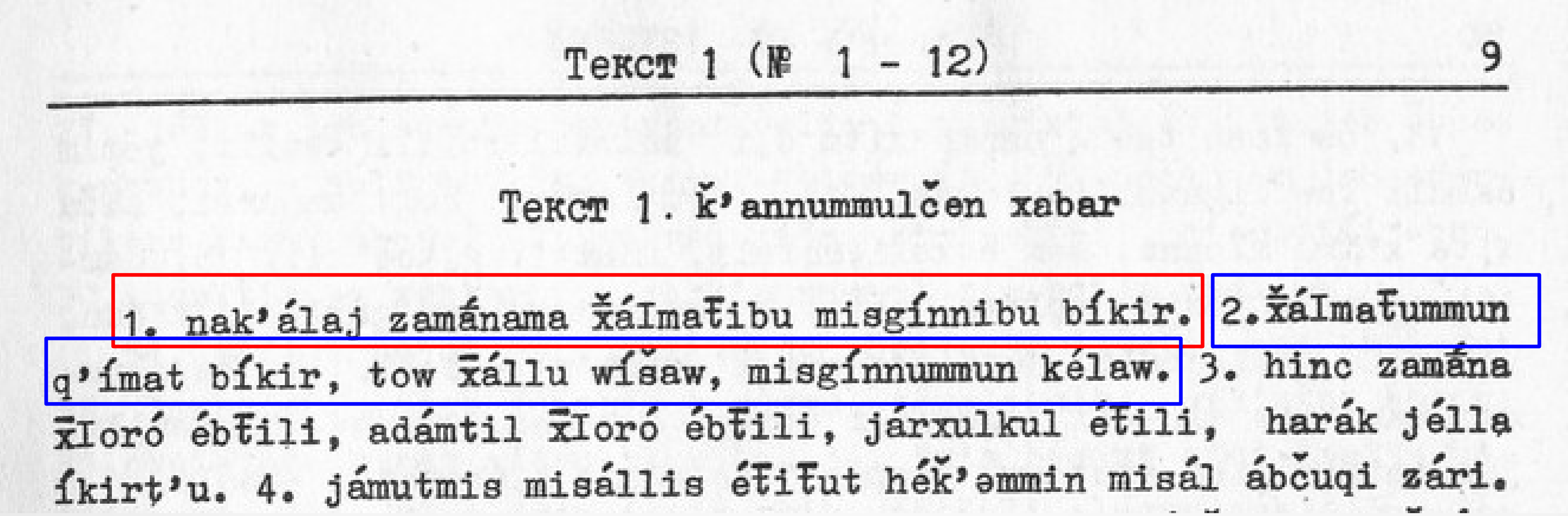
III Axis types & units: Transcribed text

Timeline (ms, digital samples)



III Axis types & units: Document

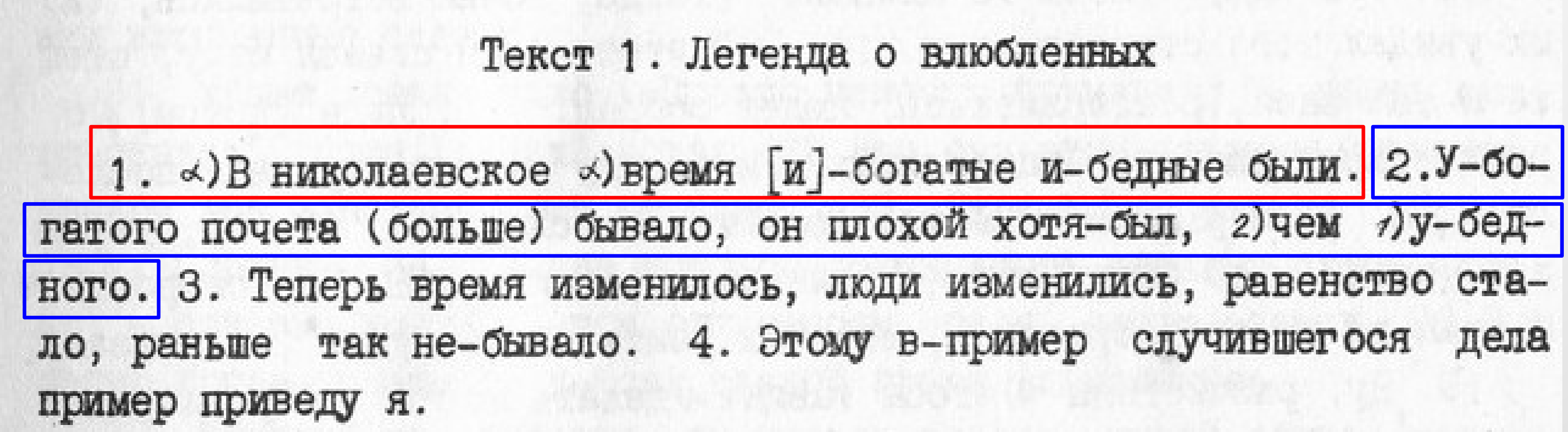
Graphic media (page + area)



Текст 1 (№ 1 - 12) 9

Текст 1. k̆'annummulčen xabar

1. nak'álaj zamā́nama x̌ álmat̄ibu misgínnibu bíkir. 2. x̌ álmat̄ummun q'ímat bíkir, tow x̄állu wíšaw, misgínnummun kélaw. 3. hinc zamā́na x̄Ioró ébt̄ili, adámtil x̄Ioró ébt̄ili, járxulkul ét̄ili, harák jélla íkirț'u. 4. jámutmis misállis ét̄it̄ut hék̆'əmmin misál ábčuqi zári.



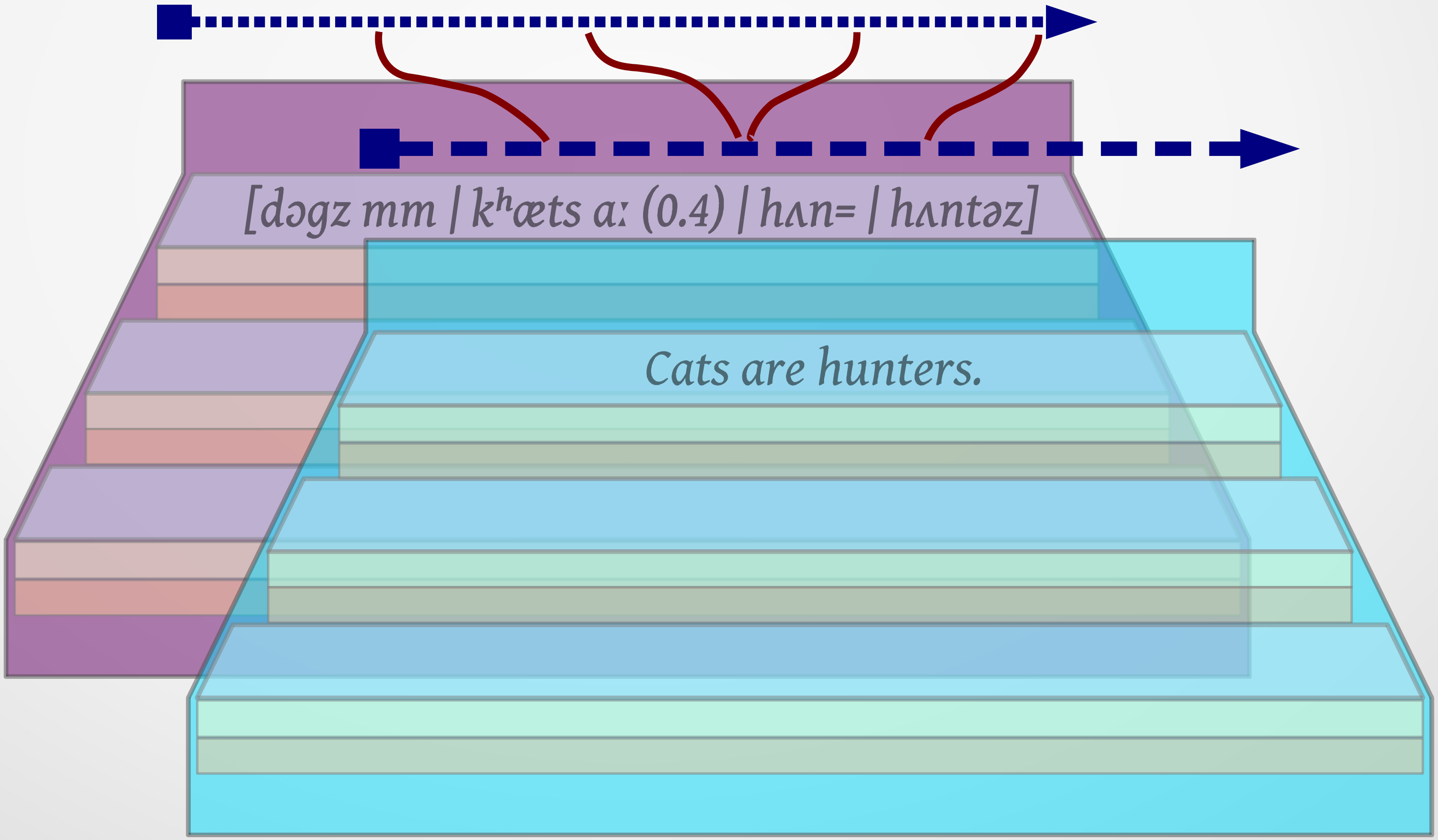
Текст 1. Легенда о влюбленных

1. α)В николаевское α)время [и]-богатые и-бедные были. 2. У-богатого почета (больше) бывало, он плохой хотя-был, 2)чем 1)у-бедного. 3. Теперь время изменилось, люди изменились, равенство стало, раньше так не-бывало. 4. Этому в-пример случившегося дела пример приведу я.

III Challenges (1): Multiple axes

Inter-axis alignment

Annotations in one axis aligned to (annotations in) another axis



III Challenges (2): Multiple speakers, languages, custom tiers, annotators...

- Multi-speaker texts are easily accounted for by introducing a "participant" attribute on segments
- Multi-lingual texts are easily accounted for by introducing a "language" attribute on segments
  - Indispensable for correctly dealing with code-switching, also quotations, borrowings etc.
  - Texts need to be separated from grammars & lexica
- Unlimited custom tiers!
- Team work: need support for comments for any object type, versioning, time/person-stamping changes

III Challenges (3): Multiple analyses

Alternative analyses of all kinds, including root annotations (e.g. alternative transcriptions; lexical and morphological homonymy; syntactic ambiguity) need to be stored and displayed as such until ambiguity is resolved

- Each alternative creates a divergence point (alternative subtrees)
- Support for feature-labeling of alternatives
  Marking divergence points for user-specified «features» allows to select for review e.g. all *open*/*close* vowel alternatives, or all *Perfect* vs. *Evidential* alternatives in a corpus
- «Feature values» for consistent choice of alternatives
  Marking each subtree for the particular analysis choice yielding this subtree allows to simultaneously settle e.g. all *open/close* vowel alternatives to *close* in one action

III Challenges (4): Non-linear markup

The basic interlinear setup is designed principally for morphological annotation, most importantly for linear annotation.

A more general format must allow for **non-linear** kinds of markup as well (e.g. dependency trees, constituency trees) necessary for full-scale syntactic or semantic analysis.

- **Groups** of (non-)contiguous annotations: multi-word expressions (e.g. periphrastic forms, idioms etc.)
- **Annotations as relations between annotations**, overlaid upon the «basic» interlinear tree

III Challenges (5): RDF and LLOD

A fully-detailed XML implementation is possible but extremely complex. Moreover, for any particular editing / management / analysis application only a part of the whole data structure would probably be relevant.

Thus one can envisage using different data formats for different purposes, cf. S. Moran's PHOIBLE project [Moran 2012; www.phoible.org] (relational DB + huge flat plain text file + RDF/OWL repository).

RDF is also a natural solution in the LLOD perspective (Linguistic Linked Open Data, see [Chiarcos et al. 2011]).

III Challenges (6): Dynamic annotations

- Analogy with Excel: seemingly simple tool is actually very powerful thanks to formula engine
- Introducing **references & formulas** into annotations will boost up research efficiency, eventually facilitating challenges (1)-(5)
- Marking up the data => getting new data

IV Dynamic annotations (1): references

Reference to another annotation

- anaphora

  I saw *Daniel*. ***He*** was running across the street...

  <item type="anaphoric" formula="{//word[@id='235']}" />

- agreement

  une **(f)** ***belle*** **(f)** *maison* (f) ‘a beautiful house’

  <item type="agr-target" formula="{//word[@id='296']}" />

- What happens with annotation identities when the text is edited? See discussion below

IV Dynamic annotations (2): lookup

Lookup expressions

- lookup part of speech, gloss etc. in lexicon

  maison => noun; feminine; ‘house’

  <item type="pos"
  formula="{$lexicon//entry[lemma=current()/../lemma]/pos}">
  noun</item>

  <item type="gls"
  formula="{$lexicon//entry[lemma=current()/../lemma]/gloss}"
  >**house**</item>

- This is actually what FLEx does but in a non configurable way: one cannot output a line with e.g. nominal gender

IV Dynamic annotations (3): functions

Numeric expressions and functions

- count words (morphemes, syllables)
- calculate tone rise/fall in semitones
- calculate distance to anaphoric target in words

String expressions and functions

- calculate CV pattern from transcription
- replace all-capitals with small caps
- convert transcription into/from IPA
- convert latin orthography into/from cyrillic

IV Dynamic annotations (4): iterations

Expressions creating multiple annotations

- (i) tokenize (text into sentences, sentences into words...) *in a configurable way!*

Iterations (loops) over multiple annotations

- (ii) for each word in given tier, lookup its pos, gloss etc. in lexicon
- combine (i) and (ii)

IV Dynamic annotations (5): How?

How to code?

- XQuery+XPath is a good candidate
- Powerful, quite compact; supports update
- Natively supported by XML databases

How to store?

- ? formulas in application only, store value (literal content)
- more preferable: store both formula and value,
user controls recalculations (lock/unlock/preview)
- what if formula generates a group of annotations?
(formula for group and values for each)

IV Dynamic annotations (6): identities

How to merge data
from different applications?

- E.g. time-align in ELAN |
  > gloss in FLEx || update alignment in ELAN
  > merge
- Merge must rely on annotation identity (e.g. GUIDs): e.g. update time for the **same sentence** (having same GUID in FLEx data as in ELAN)

IV Dynamic annotations (6): identities

What is «the same annotation»?

- Annotation properties
  - belongs to a linguistic unit (usu. "text", but maybe citation form of a sentence, word, etc.)
  - belongs to certain axis and tier
  - has position and/or parent or prev/next annotation
  - has creation attributes (annotator, timestamp)
  - has value (literal content)
  - can have complex value (formula + literal content)

IV Dynamic annotations (6): identities

Changes to which properties affect identity?

- linguistic unit => YES
- axis and tier => YES
- creation attributes (annotator, timestamp) => YES
- literal content => UNCLEAR, inform user?
(major vs. minor edits; «qualified edits»?)
- formula => probably YES
- same formula evaluated to new value => UNCLEAR, inform user?

IV Dynamic annotations (6): identities

Changes to which properties affect identity?

- parent annotation reference => YES
- parent annotation value => UNCLEAR
- previous/next annotation reference => UNCLEAR
- position on axis => UNCLEAR
  (changed one border? shifted all annotations?)

Track version for each annotation?

- Add revision attributes (annotator, timestamp, version)
- In this case, merge will be possible with updated versions of the «same» annotations, but user should be warned

V Outlook: Greater ToDo

- Fnd programmers and permanent funding :-)
- Create samples of full interlinear format
- Test different query types in eXist vs. BaseX
- Can we manage interlinear entirely in a XML database + webapp?
- Other applications:
  - dynamic metadata manager
  - registry of linguistic fieldwork (&data)
  - configurable web-publishing for texts and lexica

References

BBH 2003 — Cathy Bow, Baden Hughes and Steven Bird. 2003. Towards a General Model of Interlinear Text.

Moran 2012 — Steven Moran. 2012. Phonetics Information Base and Lexicon. Ph.D., U. of Washington.

Chiarcos et al. 2011 — Christian Chiarcos, Sebastian Hellmann, Sebastian Nordhoff. 2011. Towards a Linguistic Linked Open Data cloud: The Open Linguistics Working Group // TAL v.52 no.3, pp. 245-275.

Thank you!