Developing infrastructure for interoperating digital language archives

Gary F. Simons
SIL International

CoRSAL Symposium, UNT, Denton, TX, 17 Nov 2017

Abstract

- The digital language archiving enterprise is facing serious bottlenecks in scaling up the submission of new materials and the use of already archived materials. This talk explores the strategies of *separation of concerns* and *automation of services* in developing an infrastructure for interoperation that can break these bottlenecks.

Overview

1. What are the problems that the digital language archiving enterprise is trying to solve?

2. To solve these problems, we need an ecosystem based on “separation of concerns”

3. To bring the solution to global scale, we must maximize the automation of services

1. What are the problems we are trying to solve?

A global quest for language riches

- Document the riches of every individual language before it falls silent to the pressures of language shift
  - The collection problem
- Leverage this documentation to help shifting language communities restore the riches of their heritage
  - The revitalization problem
- Amass the existing riches of individual languages so as to mine them for new riches of linguistic insight
  - The cross-language comparison problem

The promise of technology

- Digital language documentation and description on the platform of the Web should be able to facilitate this
  - Multiple physical media are now reduced to a single digital carrier with virtually unlimited shelf space
  - Costs of creating and storing material is vastly reduced
  - Instant access to incredible amounts of information
  - Access by anyone from anywhere in the world
  - Potential for anyone in the crowd to be a producer
- Digital technologies hold the promise of language riches for everyone on a global scale

But there are some gotchas

- Riches are lost as media degrade and relentless innovation causes premature obsolescence
  - The preservation problem
- Riches are as good as lost if the people who could use them don’t know they exist or can’t find them
  - The discovery problem
- Riches lose their value when they are not available in a form that meets the user’s purpose
  - The interoperation problem

The submission bottleneck

- The vast majority of field recordings remain unarchived (and thus are at risk of loss)
- Many things hold linguists back from submitting:
  - "I will have to learn how to do archiving."
  - "It will be a lot of work to organize everything and add the metadata."
  - "First I need to do more transcription and annotation before it is ready."
- And so the archiving of recordings gets put off until a better time in the future—which may never come

The annotation bottleneck

- If a recording is archived with metadata, it can at least be preserved and discovered
- But to be used for revitalization and cross-linguistic comparison it also needs various kinds of annotation:
  - Transcription, Translation, Description of total context
  - Interlinear glossing, Structural analysis
- The collection problem is a huge one, but this one is an order of magnitude larger
  - Once we break the submission bottleneck,
    the annotation bottleneck awaits

The standardization bottleneck

- Addressing revitalization and cross-language comparison on a global scale requires interoperation at that scale
  - Interoperation occurs when information produced by one system is satisfactorily used by a different system
- But there is not global uniformity of practice—there are too many formats and conventions—and experience indicates that this is not likely to change
  - To achieve interoperation, we face another bottleneck—the bottleneck of standardizing information resources

Toward a solution

- These problems and bottlenecks are too huge to be solved by a monolithic system
  - Rather, we need an infrastructure of interoperating archives and services
- That infrastructure should form an ecosystem in which each individual system fills a distinct niche
  - Based on the principle of “separation of concerns”
- And the coverage can grow to global scale
  - By leveraging the automation of services

2. To solve these problems we need an ecosystem based on “separation of concerns”

Separation of concerns

- A long-held best practice in software engineering
  - Produces modular software that is maximally robust and maintainable under requirements for change
  - At a service level, “What belongs in my service versus what should I get from another service?”
- Concept originated with Edsgar Dijkstra (of “Go To Statement Considered Harmful” fame) in
  - 1974 essay, “On the role of scientific thought”; see full text online

Dijkstra on “separation of concerns”

“Let me try to explain to you, what to my taste is characteristic for all intelligent thinking. It is, that one is willing to study in depth an aspect of one's subject matter in isolation for the sake of its own consistency, all the time knowing that one is occupying oneself only with one of the aspects. … [N]othing is gained … by tackling these various aspects simultaneously. It is what I sometimes have called ‘the **separation of concerns**’, which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of. This is what I mean by ‘focusing one's attention upon some aspect’: it does not mean ignoring the other aspects, it is just doing justice to the fact that from this aspect's point of view, the other is irrelevant.”

The key players and their concerns

Player	*Primary concern*	*Somebody else's*
Creator	Creates new language resources	Preserving resources for long-term and presenting them to users
Archive	Curates language resources for long-term preservation & access	Creating resources and presenting them in useful ways
Service provider	Presents resources to users in a way that meets their needs	Creating new resources and preserving them for the long-term

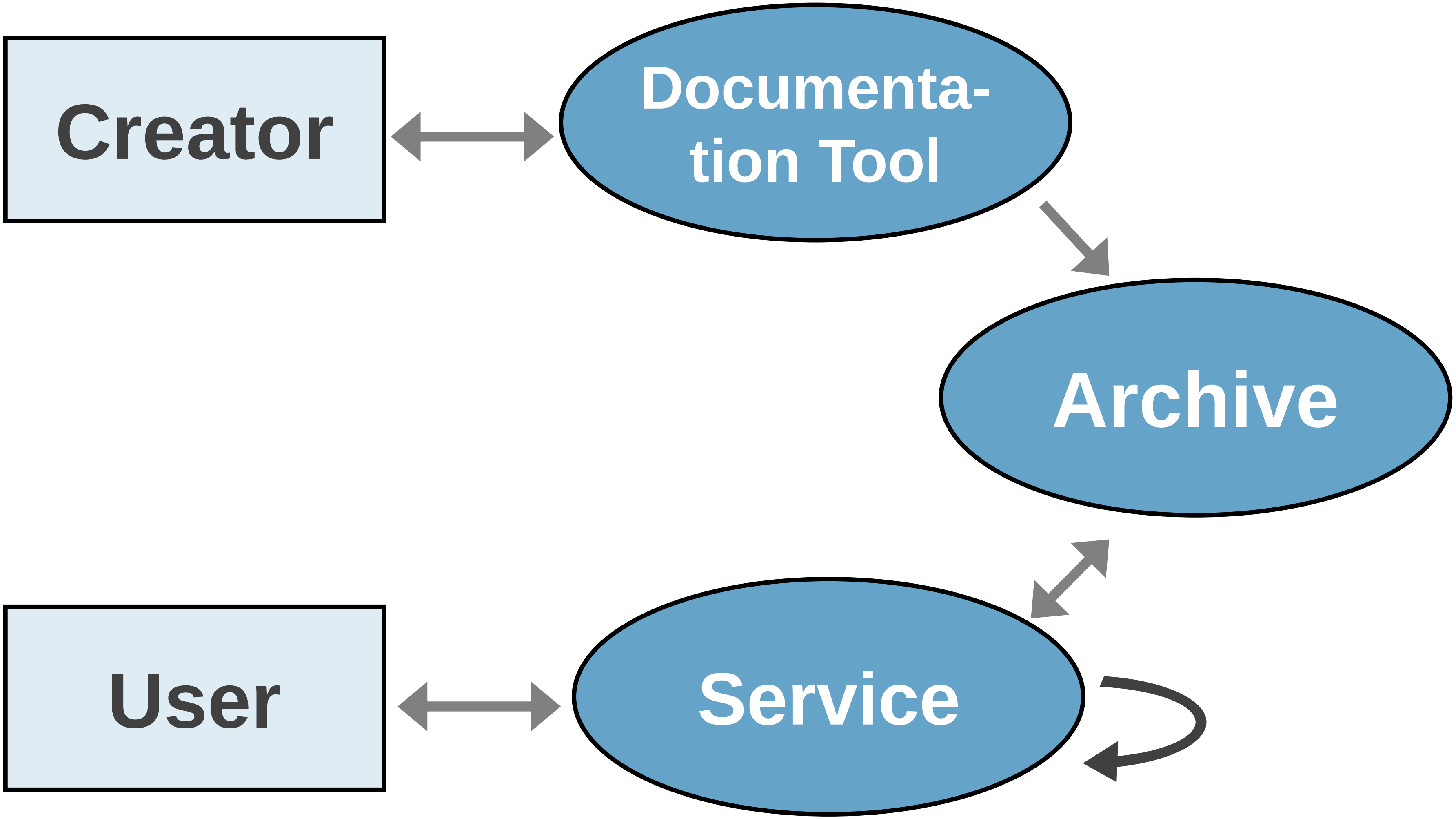
A word of caution

- If ever you are building a system for one of these concerns, and start feeling the need to address others:
  - Step away from the brink!
  - A monolithic system that addresses multiple concerns will not be sustainable
  - Instead, divide and conquer—construct a network of interoperating single-purpose systems
- A key to designing such interoperating systems is to apply separation of concerns to the information formats

Separating forms by function

From	*Function*
Working form	The form in which information is stored as it is created and edited
Presentation form	The form in which information is presented to the public
Archival form	The form in which information is stored for access long into the future
Interchange form	The form in which information is output from one system and input to another

The basic infrastructure



Documentation tools

- Resource creators use these to create language resources
- Within the tool, a working form of the information is manipulated
- The tool exports an archival form of information that provides LOTS for long-term access
  - Lossless, Open, Transparent, Suppliers
  - Descriptive XML for textual information

Archives

- Uses software like DSpace or Fedora that manages long-term preservation and access
- Ingest form for an archive is a bitstream with metadata so that it can handle any possible archival form
- Feed metadata to discovery services
- Respond to other services with requested language resources

Services

- End users interact with these to request and use language resources
- Display information to the user in a presentation form
- Can only read information in specified interchange forms
- Function is to read information into its own working form and produce the presentation form for users
- Some services allow the user to be a creator, taking input to add annotations that are then fed back to an archive so they are available to other services

3. To scale the solution we must maximize the automation of services

The scale of the problem is huge

- There are:
  - So many languages
  - So many information resources for each language
  - So many services to be provided over those resources
- That we need to automate things in order to grow to function on a global scale
  - Automating the movement: Tools to Archives to Services
  - Automating the delivery of services

Leveraging automation

Automating …	*Addresses …*
Deposit from documentation tool to archive	Submission bottleneck by removing disincentives
The things listed below	Submission bottleneck by incentivizing early submission
Annotation services	Annotation bottleneck
Translation to interchange forms	Standardization bottleneck
Presentation services	Problems of revitalization and cross-language comparison

An example for automating deposit: Language documentation at SIL International

- We have good software tools for Lang Doc and a well-used digital archive with on-line submission
  - But primary recordings are not being archived
- SIL's archive already has these incentives in place:
  - The peace of mind of long-term preservation
  - A citable "publication" that others can access
  - Management of graded access to sensitive content
- But these are eclipsed by a huge disincentive:
  - There is too much learning and work involved in turning a compiled collection into an archived corpus

The three basic tasks of Lang Doc

"Language Documentation is concerned with compiling, commenting on, and archiving language documents."
Himmelmann 1998, "Documentary and. descriptive linguistics"

1. ***Compile*** a sample of recordings of a full range of speech event types
2. ***Comment*** on those recordings
   - E.g., transcription, translation, discussion, situational context, informed consent to share
3. ***Archive*** the complete corpus of recordings and commentary with an institution that will provide long-term preservation and access

The status quo for SIL tooling

- We have a great tool for compiling and commenting
  - ***SayMore:*** *"Language Documentation Productivity"*
  - Organizes all the files and associations between them
  - Records metadata on sessions and people
  - Tracks progress on commenting workflow
  - Supports respeaking, transcription, translation
  - Download v. 3.1 at http://saymore. palaso.org/
- But it falls short of supporting the entire enterprise
  - Users are on their own to figure out how to archive their whole collection

The solution

- Automating deposit involves both preparation of the submission package and intake into the archive
  - Enhance SayMore to create an archive submission package
  - Use API on the digital archive to automate ingest
- The value proposition to the linguist should be:
  - “You can archive your corpus at the push of a button!”
- Requirements:
  - A single command causes a SayMore project to be packaged as a corpus and submitted to the archive
  - The archive submission package is known to be complete and well-formed

Reasons for returning a submission

- The metadata for the project, the sessions, or the participants is incomplete
- There is no introductory document describing the project and its methods
- There are no “Table of contents” documents listing all the sessions and all the participants
- There are participants who have not given consent for public identification and have not been anonymized
- There are materials marked for release to the public that lack informed consent to share
- There are files not attributed to any participants or in formats that are not accepted by the archive

Automating quality checking

- These are conditions that software can detect
  - So automate them! Then linguists can be alerted and fix them long before submitting to the archive
- Thus we need to add a “Preflight for archiving” function:
  - Warns of a missing Introduction
  - Identifies every missing obligatory metadata element
  - Identifies every file that is not attributed to any participant
  - Identifies every file in a format not accepted by the archive
  - Identifies every session marked for public release that is missing informed consent to share

Specifications for “Archive now” button

- Update the automatically generated “tables of contents”
- Generate and insert the “preflight” report for the curator
- Organize the sessions into collections by access level and anonymize as needed
- Place the key to anonymization in a curators-only folder
- Generate the corpus metadata record as a METS package
- Bundle the corpus contents into bitstreams that are ZIP files of up to 1 Gigabyte each
- Use SWORD API on the DSpace repository to automate submission of the METS package and all the bitstreams

Automating annotation services

- Status quo (cf. AARDVARC project)
  - Linguists perceive completion of transcription (and other annotation) as a prerequisite for archiving
  - Linguists typically attack this problem by themselves
  - They do not use state-of-the-art automated annotation tools since they aren’t easily installed
    - speech activity detection
    - speaker diarization (*i.e.*, segmenting into turns with speaker id)
    - automatic transcription of oral translations in major languages
    - machine learning of models for language-specific annotation

Automating annotation services

- Envisioned future
  - Archives provide for processing of deposited materials with state-of-the-art automated annotation tools
  - Thus an immediate benefit of depositing in an archive is access to these automated annotation tools
  - Archive deposits should be progressively enriched via stand-off annotations attributed to the annotator (who could be someone other than the submitter) so that absence of annotation need no longer delay archiving

An example:
The Language Application Grid

- An NSF grant project (http://www.lappsgrid.org/)
  - The Language Application Grid: A Framework for Rapid Adaptation and Reuse
  - Vassar, Brandeis, CMU, Linguistic Data Consortium
- The Grid consists of nodes on the Internet providing:
  - Data services—Provide access to archived corpora
  - Processing services—Provide access to natural language processing (NLP) tools
  - Composition of services—Create workflows to run data through one or more processes
- An archive could provide services over its deposited material by sending it to processing services elsewhere

Mobilizing the crowd

- Building an archive is one thing
  - Filling it with recordings that have a full complement of annotations is quite another
- Realizing the vision of documenting the riches of every language is going to require that we
  - Mobilize the research community to participate
  - Mobilize speaker communities to participate
  - Mobilize citizen scientists to participate
- Our infrastructure needs to support collaboration among all these players on a global scale

Resources as open-ended

- Archives should support an open-ended annotation process in which an annotation submission uses stand-off markup and its own metadata record to link to what it annotates
- After a recording is deposited, other players could
  - Add careful respeaking
  - Add a translation (either oral or written)
  - Add a transcription (of text or of translation)
  - Add a translation of the translation to yet another language
  - Add POS tagging or other grammatical analysis
  - Invoke an automatic process for any of the above and revise the automatically produced result

Automating workflow

- The types of the annotations already associated with a resource (and the languages they are in) comprise the state of that resource
- An annotation task is an operation on that state
    - Each annotation task has a prerequisite state
    - Performing the task changes the state of the resource
- This defines an implicit workflow that can be automated
    - For any resource, there is a set of possible next tasks
    - The infrastructure needs to manage that workflow

Matching supply and demand

- We need to match up two things:
  - The huge demand for annotation tasks to be done — all of the possible next tasks for all resources
  - The supply of people worldwide who could do them
- The infrastructure needs to include a marketplace that matches supply with demand
  - *E.g.*, eBay, eHarmony, mTurk.com
- Match against a user's skill profile to find next tasks to do
  - E.g., TED's Open Translation Project using Amara
    - Web tool to segment videos and add subtitles
    - 29,000 translators → 120,000 translations in 115 languages

Providing end-user services

- One-off custom service for a single language
  - We can’t afford many of these!
- Multitenant service
  - Many tenants supported by the same installed code
  - Presentation is automated from common interchange form
  - E.g., webonary.org (>100 dictionaries sourced from FLEx)
- Aggregation service
  - Provides comparison and search across languages
  - Data are comparable because of common interchange form
  - E.g., OLAC (interoperable search over 60 archives)

The rule of separation

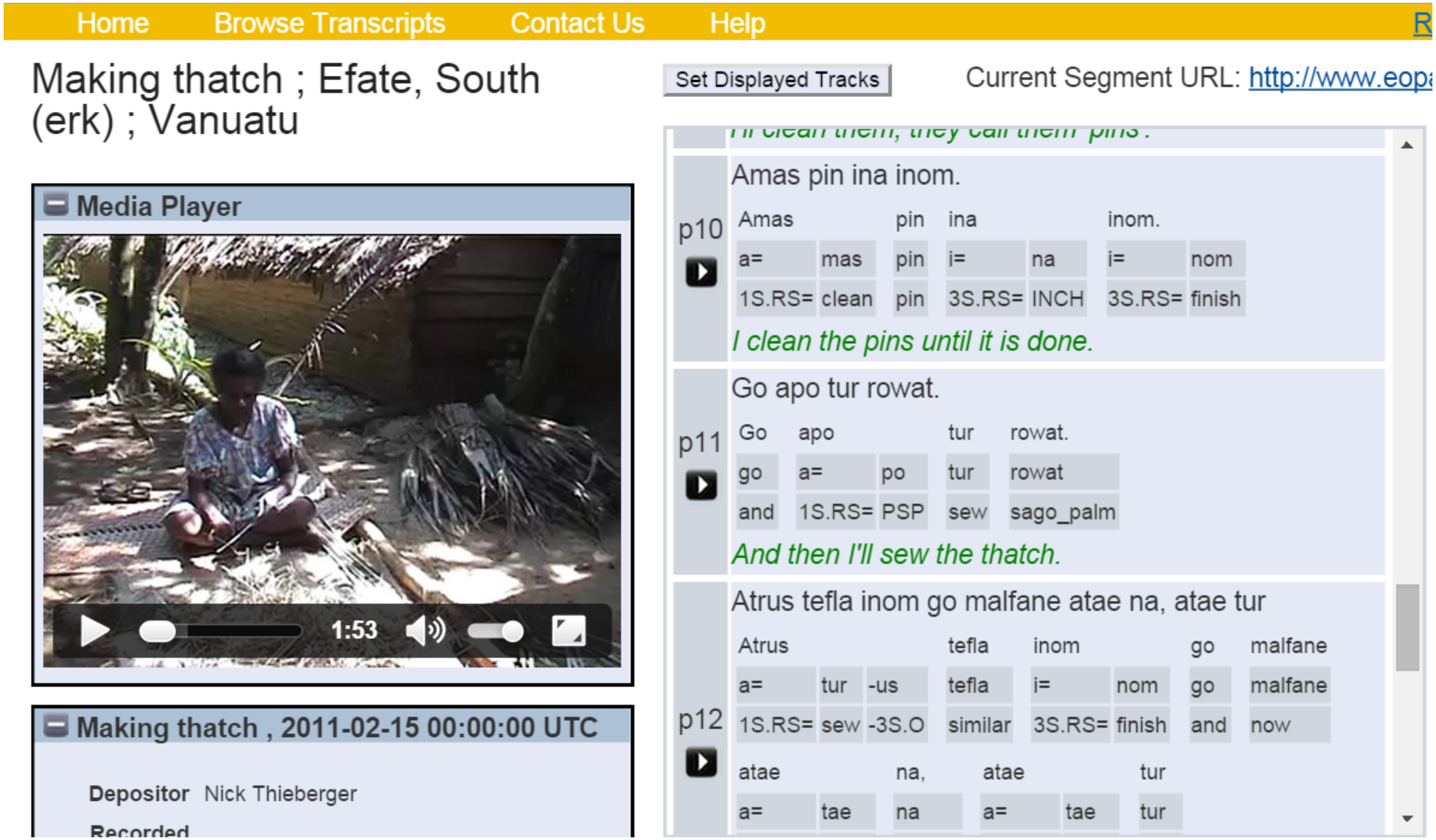
- Maintaining the separation of concerns demands that:
  - There can be no language-specific programming inside a multitenant service or an aggregation service
- We must
  - Generalize behaviors into features that are relevant to multiple languages and give them a representation in the interchange form
  - Place language-specific programming in the translation from the working form of the language data to the interchange form accepted by the service

Automating standardization

- We achieve standardization for interoperation
  - Not by getting people to adopt a single practice
  - But by automatically translating from the form they have produced to the interchange form that is needed
- As long as there is a handful of common archival forms it is not overly burdensome to build translators
  - Users will gravitate toward archival forms that are widely supported by services since they will see the rewards
  - Totally idiosyncratic behavior will be unrewarded

An example of automated presentation

- EOPAS: Ethnographic E-Research Online Presentation System, *from School of Language and Linguistics, University of Melbourne*



Making thatch ; Efate, South (erk) ; Vanuatu

Current Segment URL: http://www.eopa

Making thatch , 2011-02-15 00:00:00 UTC

Depositor Nick Thieberger

Recorded

I'll clean them, they call them 'pins'.

p10

Amas pin ina inom.

Amas		pin	ina		inom.	
a=	mas	pin	i=	na	i=	nom
1S.RS=	clean	pin	3S.RS=	INCH	3S.RS=	finish

I clean the pins until it is done.

p11

Go apo tur rowat.

Go	apo		tur	rowat.
go	a=	po	tur	rowat
and	1S.RS=	PSP	sew	sago_palm

And then I'll sew the thatch.

p12

Atrus tefla inom go malfane atae na, atae tur

Atrus			tefla	inom		go	malfane
a=	tur	-us	tefla	i=	nom	go	malfane
1S.RS=	sew	-3S.O	similar	3S.RS=	finish	and	now

atae		na,	atae		tur
a=	tae	na	a=	tae	tur

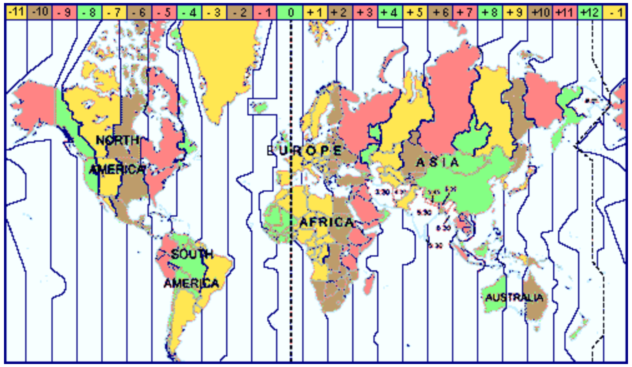
Input to EOPAS

- EOPAS defines its own interchange form:
  - EOPAS XML format
- It provides XSLT scripts to translate from:
  - Toolbox XML format
  - Transcriber XML format
  - ELAN XML format
- And more could be developed:
  - Xigt, FLExText, TEI

The problem of semantic interoperation

- Syntactic interoperation (via a common XML format) is relatively straightforward and adequate for multitenant services (since the tenants are isolated)
- But an aggregation service also wants semantic interoperation to maximize benefit, *e.g.,*
  - OLAC controlled vocabularies: Language Identification (ISO 639-3), Linguistic Data Type, Linguistic Field of Study, Participant Role, Discourse Type
  - GOLD: General Ontology for Linguistic Description

A helpful metaphor

- Using cross-language comparison for finding similarities is more broadly useful than for finding equivalencies
- For cross-linguistic search, we need to tell time not by the exactness of solar noon but by the utility of time zones
- 

(source)
- Mapping disparate data to equivalent points is very hard; but translating to the right zone is much easier and serves our purpose of searching for things in the same zone

Automating a global service

- Begin by automating discovery
  - Use OLAC to discover new resources in a known format
    - N.B. This is possible in theory, but data providers will need to start using a standard vocabulary to name the formats and use a standard like OAI-ORE to get inside corpora
- Run the correct transformation script to translate from the source format to the interchange format
- Load the new data into the service
- Expand the reach by implementing more translators for more source formats

Conclusion

- So what's the future of digital language archiving?
  - Automation!
- It holds the key to the transition from archives being:
  - The inadequately populated final resting place for long-term preservation of potentially valuable material
- To becoming:
  - An overflowing storehouse of global language riches that are meeting the needs of real users

To fill the storehouse, we must …

- Remove disincentives for submission by
  - Automating quality checking
  - Automating the submission process
- Provide incentives for submission by
  - Automating annotation services
  - Automating crowd workflow
- And feeding archived resources to end-user services by
  - Automating translation to standardized interchange forms
  - Automating creation of presentation forms