TEI and Git in ParlaMint: Collaborative Development of Language Resources

Tomaž Erjavec
Dept. of Knowledge Technologies
Jožef Stefan Institute
Ljubljana, Slovenia
tomaz.erjavec@ijs.si

Matyáš Kopp
Faculty of Mathematics and Physics
Charles University
Prague, Czech Republic
kopp@ufal.mff.cuni.cz

CLARIN Annual Conference
October 10–12, 2022

Overview of the talk

Introduction

The ParlaMint projects

- CLARIN ERIC supported projects; joint effort of many partners
- Centers around compiling a set of comparable, richly annotated corpora of parliamentary debates in Europe
- ParlaMint I (2020–2021): 17 parliaments, 500M words, 11,000 speakers, structurally & linguistically annotated
  - focus on interoperability of encoding and data
- ParlaMint II (2022–2023): extend time period + 12 new (also regional) parliaments
  - focus on metadata structure, compatibiliy and extension
  - also MT to English and speech data for a subset.

Countries and regions

Prerequisites of compiling the corpora

The number of corpora and the richness of encoding means it is important to have:

- robust but easily maintainable encoding, along with documentation
- automated validation and conversion procedures for the corpora
- support for collaborative development with versioning, attribution and comparisons of files

Text Encoding Initiative

TEI and ParlaMint

- The TEI covers all types of encoding in ParlaMint corpora
- Encoding is in XML, to our customisation of the TEI Guidelines
- A TEI customisation is specified in a TEI ODD document
- ODD contains both the prose encoding guidelines and the formal schema of the customisation
- With TEI XSLT stylesheets ODD is converted to:
  1. ODD prose guidelines to HTML for reading,
  2. ODD schema to XML schema languages (e.g. RelaxNG)
- The reality is somewhat more complicated:
  1. ParlaFormat workshop: Parla-CLARIN ODD
  2. ParlaMint I: ParlaMint RelaxNG
  3. ParlaMint II: ParlaMint ODD

Validation in ParlaMint

Five stages in validation:

1. Formal XML validation is performed with the ParlaMint RelaxNG schema
2. Further validation (links, content, relations betwen organisations and persons) is performed by the XSLT scripts
3. Other XSLT scripts converted XML to downstream formats: conversion scripts can expose furhter errors
4. ParlaMint is mounted on concordancers: analysis of corpora reveals bugs in the data
5. Eagle-eye validation

As a result of this formal and functional validation corpora are less noisy and more interoperable.

Git and GitHub

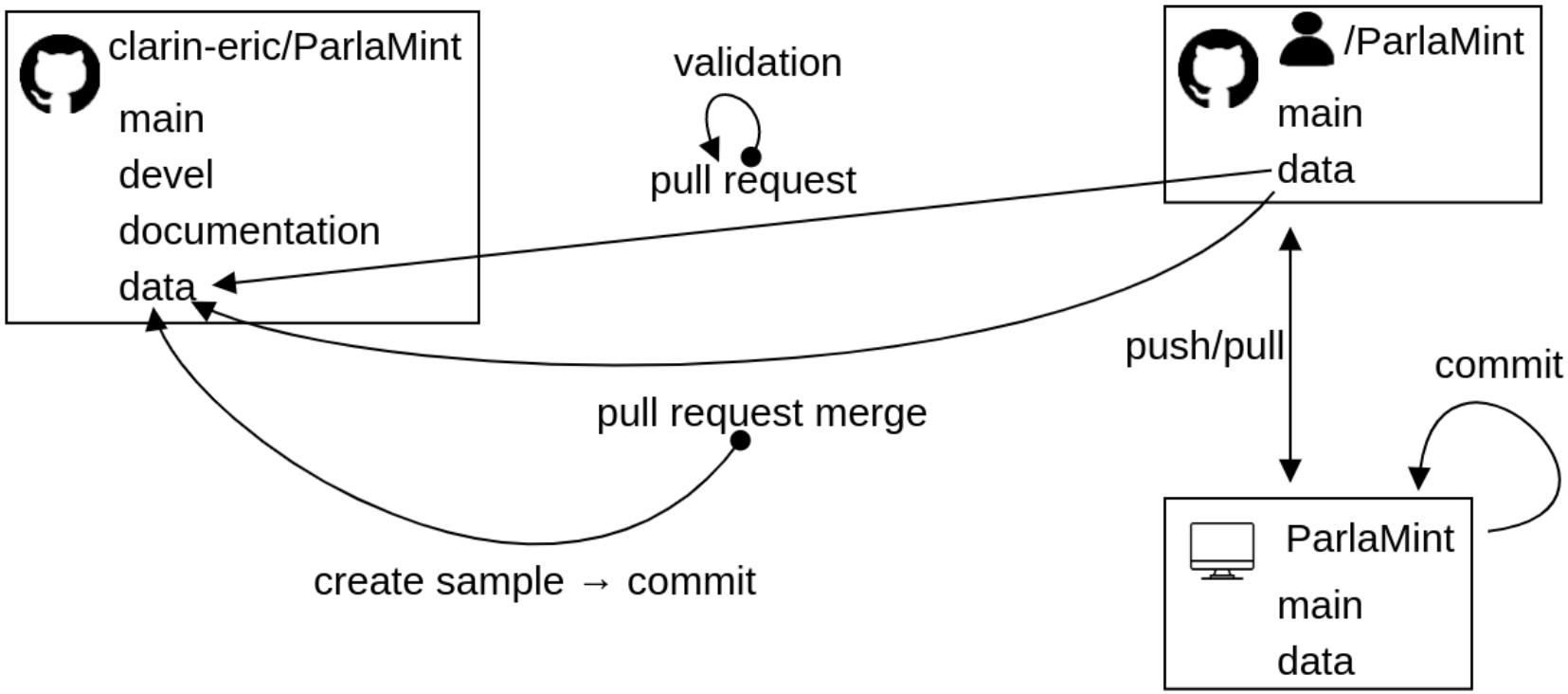
ParlaMint and Git(Hub)

- Git: distributed revision control system for software development
- GitHub: Git hosting platform with further functionality (issues, pages, actions)
- Git(Hub) also used for collaborative development of LRs (Universal Dependencies, ELTeC) and TEI customisations (Lex-0, ELTeC, Parla-CLARIN)
- Although ParlaMint too large to be fully Git-based, the development environment is:
  guidelines, schema, scripts, complete metadata, samples of data, also in derived formats

Development process

- GitHub issues used for reporting and documenting problems
- GitHub pages used for publishing the ParlaMint encoding guidelines
- New data samples added or revised with GitHub pull requests
- Pull requests trigger validation with GitHub actions
- Publishable samples and derived formats also made with GitHub actions
- Local validation of a complete corpus:
  the self-documenting Unix `make` checks prerequisites, validates a corpus, converts to derived formats

Development process



Conclusions

Conclusions

- TEI can be used to specify the encoding documentation & schema for language corpora (or other types of LRs)
- Git is well suited for controlled & distributed development (& publishing) of LRs and encoding guidelines & schemas
- We believe that TEI & especially Git are not as well known in the SSH community as they should be:
  adopting them into the work process could go a long way in making the (esp. collaborative) development of LRs a much smoother and more controlled process
- Warning: fully mastering TEI and Git is complicated, and (CLARIN?) tutorials could be welcome for SSH scholars

Further work

- Continue working on the ParlaMint schema and guidelines: new corpora, annotation and resource types
- Continue working on the Git(Hub) environment
- Decentralise the development of the ParlaMint corpora: anyone wishing to produce a ParlaMint-compatible corpus should be able to do so independently

TEI and Git in ParlaMint: Collaborative Development of Language Resources

Tomaž Erjavec
Dept. of Knowledge Technologies
Jožef Stefan Institute
Ljubljana, Slovenia
tomaz.erjavec@ijs.si

Matyáš Kopp
Faculty of Mathematics and Physics
Charles University
Prague, Czech Republic
kopp@ufal.mff.cuni.cz

CLARIN Annual Conference
October 10–12, 2022