

Encoding and Designing for the Swift Poems Project

Jonathan Swift and the Text Encoding Initiative

James R. Griffin III
Digital Library Developer
Lafayette College Libraries

Introductions



James Woolley

(Emeritus) Frank Lee and Edna M. Smith Professor of English at Lafayette College



Stephen Karian

Associate Professor of English at the University of Missouri



James R. Griffin III

Digital Library Developer at the Lafayette College Libraries

Overview of the Swift Poems Project

- Woolley and Karian seek to archive poems attributed to Jonathan Swift
- Beginning in 1987, this has involved:
 - Identifying and Cataloging Primary Sources
 - Transcription
 - Copy-typing, encoding, and annotating the primary sources
 - This method has **not** relied upon the usage of the TEI Collation
 - Collation
 - Identifying the copy-text (and variant texts) for any given poem

Overview of the Swift Poems Project

- The Libraries at Lafayette
 - In 2009 Woolley consulted with the Libraries for assistance with the Project
 - Visual Resources Curator (Paul Miller) developed a set of Microsoft Access Databases
 - These structure the catalogs maintained by Woolley and Karian
- In 2012, the NEH awarded a Scholarly Editions Grant for the Project
 - This includes supporting the development of a digital edition
 - This will be integrated into volumes of the *Cambridge Edition of the Works of Jonathan Swift*
 - Digital Scholarship Services (Department in the Libraries)
 - Agreed to support this project formally using a planned migration to Fedora Commons
 - Griffin joined Digital Scholarship Services in the role of digital library developer

Identifying and Cataloging Primary Sources

- Identifying the Sources

- Sources are 18th century printed and manuscript texts
 - To date, over 6500 manuscripts have been identified and cataloged
- Few digital surrogates for the printed and manuscript texts are available
 - Of these only a restricted set aren't protected under copyright

- Cataloging the Sources

- Bibliographic metadata are structured for each source
 - Elements are extracted from external catalogs (e. g. English Short Title Catalogue)
- An author attribution may be specified (but it is rarely authoritative)
- Not all sources are cataloged with an authoritative title
 - **An internal identifier is used to reference poems as a result**

Transcribing the Primary Sources

640-35D- 41 As he whom «MDUL»Ph\oe\bus«MDNM» in his Ire
640-35D- 42 Hath «MDUL»blasted«MDNM» with Poetick Fire.
640-35D- 43 IWhat Hope of Custom in the «MDUL»Fair«MDNM»,
640-35D- 44 While not a Soul demands your Ware?
640-35D- 45 Where you have nothing to produce
640-35D- 46 For private Life, or publick Use?
640-35D- 47 «MDUL»Court, City, Country«MDNM» want you not;
640-35D- 48 You cannot bribe, betray, or plot.
640-35D- 49 For Poets Law makes no Provision:
640-35D- 50 The Wealthy have you in Derision.
640-35D- 51 Of State-Affairs you cannot smatter;
640-35D- 52 Are awkward when you try to flatter.
640-35D- 53 Your Portion, taking «MDUL»Britain«MDNM» round,
640-35D- 54 «FNI.«MDUL»Paid to the Poet Laureat, which Place was given to one «MDNM»Cibber«MDUL», a Player«MDNM».)Was just one annual Hundred Pound.
640-35D- 55 Now not so much as in Remainder
640-35D- 56 Since «MDUL»Cibber«MDNM» brought in an Attainder;
640-35D- 57 Nor ever fixt by Right Divine
640-35D- 58 (A Monarch's Right) on «MDUL»Grubstreet«MDNM» Line.
640-35D- 59 IPoor starvling Bard, how small thy Gains!
640-35D- 60 How unproportion'd to thy Pains!
640-35D- 61 And here a «MDUL»Simile«MDNM» comes pat in:
640-35D- 62 Though «MDUL»Chickens«MDNM» take a Week to fatten,
640-35D- 63 The Guests in less than half an Hour
640-35D- 64 Will more than half a Score devour.
640-35D- 65 So, after toiling twenty Days,
640-35D- 66 To earn a Stock of Pence and Praise,
640-35D- 67 Thy Labours grown the Critick's Prey,
640-35D- 68 Are swallow'd o'er a Dish of Tea;
640-35D- 69 Gone, to be never heard of more;
640-35D- 70 Gone, where the «MDUL»Chickens«MDNM» went before.
640-35D- 71 IHow shall a new Attempter learn
640-35D- 72 «MDUL»Of diff'rent Spirits to discern«MDNM».
640-35D- 73 And how distinguish, which is which,
640-35D- 74 The Poet's Vein or scribbling Itch?
640-35D- 75 Then hear an old experienc'd Sinner
640-35D- 76 Instructing thus a young Beginner.

Transcribing the Primary Sources

- The transcripts themselves are created using the *Nota Bene* application
 - *Nota Bene* encodes textual structure using a system of tags termed as “mode codes”

```
«MDUL»This mode encodes italicized style rendering«MDNM»  
«MDBO»This mode encodes black letter«MDNM»
```

- The researchers have further extended this system to support editorial annotation:

```
Lorem\«MDUL»add·caret«MDNM»·this text added with a caret\ipsum  
  
Dolor\«MDUL»del«MDNM»·this was deleted·\sit amet
```

- Not all instances of annotative markup require mode code tags:

```
\pasted·over\  
  
\printed text\  

```

Accessing and Preserving the Transcripts

- Accessing the Nota Bene comes with challenges
 - The Nota Bene release used by the researchers has been 3.0 (released in 1988)
 - Accessing the Nota Bene directly would require a virtualized environment for Microsoft DOS
 - The Nota Bene transcripts are managed as `text/plain` media resources
- The Text Encoding Initiative P5
 - Provides a robust data model
 - Standardized and open format for interchange
 - A more effective solution for preservation

Encoding the Transcripts

File Edit Options Buffers Tools SGML Help

```
<!--42" xml:id="spp-640-350--line-42">Math
<hi rend="underline">blasted</hi> with Poetick Fire.</l>
</l>
<!--43" xml:id="spp-640-350--line-43">What Hope of Custom in the
<hi rend="underline">Fair</hi>,</l>
</l>
<!--44" xml:id="spp-640-350--line-44">While not a Soul demands your Ware?</l>
</l>
<!--45" xml:id="spp-640-350--line-45">Where you have nothing to produce</l>
</l>
<!--46" xml:id="spp-640-350--line-46">For private Life, or publick Use?</l>
</l>
<!--47" xml:id="spp-640-350--line-47">
<hi rend="underline">Court, City, Country</hi> want you not;</l>
</l>
<!--48" xml:id="spp-640-350--line-48">You cannot bribe, betray, or plot.</l>
</l>
<!--49" xml:id="spp-640-350--line-49">For Poets Law makes no Provision:</l>
</l>
<!--50" xml:id="spp-640-350--line-50">The Wealthy have you in Derision.</l>
</l>
<!--51" xml:id="spp-640-350--line-51">Of State-Affairs you cannot smatter;</l>
</l>
<!--52" xml:id="spp-640-350--line-52">Are awkward when you try to flatter.</l>
</l>
<!--53" xml:id="spp-640-350--line-53">Your Portion, taking
<hi rend="underline">Brittain</hi> round,</l>
</l>
<!--54" xml:id="spp-640-350--line-54">
<ref target="#spp-640-350--footnote-1">1</ref>
<note place="foot" n="1" xml:id="spp-640-350--footnote-1">
<hi rend="underline">Paid to the Poet Laureat, which Place was given to one </hi><hi>Cibber
<hi rend="underline">, a Player</hi></note>Was just one annual Hundred Pound.</l>
</l>
<!--55" xml:id="spp-640-350--line-55">Now not so much as in Remainder</l>
</l>
<!--56" xml:id="spp-640-350--line-56">Since
<hi rend="underline">Cibber</hi> brought in an Attainder;</l>
</l>
<!--57" xml:id="spp-640-350--line-57">Nor ever fixt by Right Divine</l>
</l>
<!--58" xml:id="spp-640-350--line-58">(A Monarch's Right) on
<hi rend="underline">Grubstreet</hi> Line.</l>
</l>
<!--59" xml:id="spp-640-350--line-59">Poor starvling Bard, how small thy Gains!</l>
</l>
<!--60" xml:id="spp-640-350--line-60">How unproportion'd to thy Pains!</l>
</l>
<!--61" xml:id="spp-640-350--line-61">And here a
<hi rend="underline">Simile</hi> comes pat in:</l>
</l>
```


UUU:***-F1 640-350--tei.xml All L1 (XML)-

On Poetry A Rhapsody(Poem 640-350-)

Encoding the Transcripts

- Encoding using the TEI-P5 could not be a manual process
 - The researchers required a system to transform *Nota Bene* into a TEI-XML implementation
 - An API for Ruby using Nokogiri was developed to support this
- Viewing the TEI-XML was of limited value
 - Research techniques driven by Nota Bene require a rendering of the text
 - Styled HTML5 (using Twitter Bootstrap) serves as a minimum viable product
 - Improvements can be rapidly prototyped for the encoding
 - This approach takes inspiration from Agile software development practices
 - The stakeholders have continuously improving (or maturing) prototypes
 - The approach is also draws upon “pair programming” within eXtreme Programming

Viewing the Encoded Transcript

 Home About

Browse by Poem ID

Search by Text

Search

42 Hath blasted with Poetick Fire.
43 What Hope of Custom in the Fair,
44 While not a Soul demands your Ware?
45 Where you have nothing to produce
46 For private Life, or publick Use?
47 Court, City, Country want you not;
48 You cannot bribe, betray, or plot.
49 For Poets Law makes no Provision:
50 The Wealthy have you in Derision.
51 Of State-Affairs you cannot smatter;
52 Are awkward when you try to flatter.
53 Your Portion, taking Britain round,
54 Was just one annual Hundred Pound.
55 Now not so much as in Remainder
56 Since Cibber brought in an Attainder;
57 Nor ever fixt by Right Divine
58 (A Monarch's Right) on Grubstreet Line.
59 Poor starvling Bard, how small thy Gains!
60 How unproportion'd to thy Pains!
61 And here a Simile comes pat in:
62 Though Chickens take a Week to fatten,
63 The Guests in less than half an Hour
64 Will more than half a Score devour.
65 So, after toiling twenty Days,
66 To earn a Stock of Pence and Praise,
67 Thy Labours grown the Critick's Prey,
68 Are swallow'd o'er a Dish of Tea;
69 Gone, to be never heard of more;
70 Gone, where the Chickens went before.
71 How shall a new Attempter learn
72 Of diff'rent Spirits to discern.
73 And how distinguish, which is which,
74 The Poet's Vein or scribbling Itch?
75 Then hear an old experienc'd Sinner
76 Instructing thus a young Beginner.

Digital Scholarship Services Skillman Library Lafayette College

On Poetry A Rapsody (Poem 640-35D-)

Enriching the Encoded Transcripts

- Limits are obviously present with this approach
 - Researchers are **not** encoding the transcripts using the TEI P5
 - The developer for the Ruby API is not a literary scholar
- How can this encoding be made collaborative?
 - The developer and the researcher could operate in a shared environment
 - This is inspired heavily by the *pair-programming* technique within eXtreme Programming
 - In this case, both the developer and a researcher share a physical working environment

Enriching the Encoded Transcripts

- Collaborative encoding and quality control
 - The researchers will identify faults in the rendered transcripts
 - The developer can extend the Ruby API, XSL, or styling for the HTML5
 - This enables rapid prototyping of the interface
 - Delivery time for the researcher (in transcribing the sources) can be increased
 - In response, the developer can more readily scope improvement requests
- Textual criticism is still not enabled by this approach
 - The researchers must identify variant readings to a given text
 - Critical apparata are not explicitly encoded within the *Nota Bene* transcripts

Collation for the Swift Poems Project

- Collation as a solution
 - Originally the researchers collated the *Nota Bene* transcripts using a *FoxPro* program
 - Visualization was used to identify variation
 - Tokenization was customized
 - A set of controlled characters (&,~,|,#) symbolized differences in structure


```
664D721L 1 The Pulteney's and Shippens & such folk
664D233Y 1 Well may Poultney & Shippen rant, grumble,
664D360Y 1 Your Pulteney & Shippen, ~ ~ folks

664D721L 2 How unlucky it is for the Nation #####
664D233Y 2 |& curse the hard Fate of the Nation;
664D360Y 2 |How hard is the Fate of the Nation
```


Collation within a Digital Scholarly Edition


- A collation interface was scoped for the digital edition
 - This interface must enable the transition from the legacy collation engine
- Collation features could be extended
 - Lines are still tokenized
 - Initially attempted to preprocess the text and use the Penn Treebank tokenizer
 - Ultimately found that abstracting the tokenizer was simply more effective
 - Alignment is addressed without the use of controlled characters
 - The edit distance between tokens can be calculated
- Experimental features can be introduced
 - Part-of-speech tagging to further enhance textual analysis
 - Currently a pretrained Perceptron tagger is being tested
 - May investigate more performant approaches (e. g. Hidden Markov Model)

Collation within a Digital Scholarly Edition





Digital
Scholarship
Services

[Home](#) [About](#)

Browse by Poem ID 

Search by Text  Search

Transcripts

		Transcript ID	Copy-Text	Variant Texts
	View	!W1908G1	<input type="checkbox"/>	<input type="checkbox"/>
	View	!W1910M1	<input type="checkbox"/>	<input type="checkbox"/>
	View	!W1910M2	<input type="checkbox"/>	<input type="checkbox"/>
	View	!W1912H1	<input type="checkbox"/>	<input type="checkbox"/>

Mode


Nota Bene

☒


Text Encoding Initiative (TEI)

☐

Tokenizer

Swift Sentence Tokenizer 

Part-of-Speech Tagging

Disabled 

Collation Output

Collate

Reset

Digital Scholarship Services

Skillman Library

Lafayette College

Collating Variants for the Poem !W190

Collation within a Digital Scholarly Edition

- The collation can also address flaws in the encoding
 - By default, all unencoded Nota Bene markup is stripped from the TEI
 - Users will be able to collate the texts and visualize differences
 - Optionally Nota Bene can be preserved
 - Researchers still retain access to some of the controlled characters
 - Researchers and the developer can identify unencoded *Nota Bene* sequences
- A heatmap is currently the supported visualization
 - This is a straightforward means of rendering the textual differences


Collation within a Digital Scholarly Edition

Collation for 640-

640-35D- (Copy-Text)


640-36L- (Variant)

640-34L2 (Variant)



Home

About



Search

640-35D-	Line 42	Hath	blasted	with	Poetick	Fire.	
640-36L-	Line 42	Hath	blasted	with	Poetick	Fire.	
640-34L2	Line 42	Hath	blasted	with	poetick	Fire.	
640-35D-	Line 43	What	Hope	of	Custom	in	the
640-36L-	Line 43	What	Hope	of	Custom	in	the
640-34L2	Line 43	What	Hope	of	Custom	in	the
640-35D-	Line 44	While	not	a	Soul	demands	your
640-36L-	Line 44	While	not	a	Soul	demands	your
640-34L2	Line 44	While	not	a	Soul	demands	your
640-35D-	Line 45	Where	you	have	nothing	to	produce
640-36L-	Line 45	Where	you	have	nothing	to	produce
640-34L2	Line 45	Where	you	have	nothing	to	produce
640-35D-	Line 46	For	private	Life,	or	publick	Use?
640-36L-	Line 46	For	private	Life,	or	publick	Use?
640-34L2	Line 46	For	private	Life,	or	publick	Use?
640-35D-	Line 47	Court,	City,	Country	want	you	not;
640-36L-	Line 47	Court,	City,	Country,	want	you	not;
640-34L2	Line 47	Court,	City,	Country,	want	you	not;
640-35D-	Line 48	You	cannot	bribe,	betray,	or	plot.
640-36L-	Line 48	You	cannot	bribe,	betray	or	plot.

Digital Scholarship Services

Skillman Library

Lafayette College

Forthcoming Features

- Design Improvements

- Stakeholders have driven the requirements for the UI
 - Interviewing and testing for public users must be undertaken
- Extending UI features using JavaScript frameworks
 - The digital edition is current implemented in the Tornado framework for Python
 - Solutions such as AngularJS and React reduce UI to a set of modular components
 - They also require a RESTful API to be implemented

- Preservation

- Ingestion of the critically edited reading texts in the TEI-XML
- Lafayette College Libraries is a member of the Project Hydra community
 - Migration for other systems (Islandora and DSpace) is underway
 - Modeling TEI resources in Hydra could then expose metadata elements in the RDF

Encoding and Designing for the Swift Poems Project

Thank you for your attention



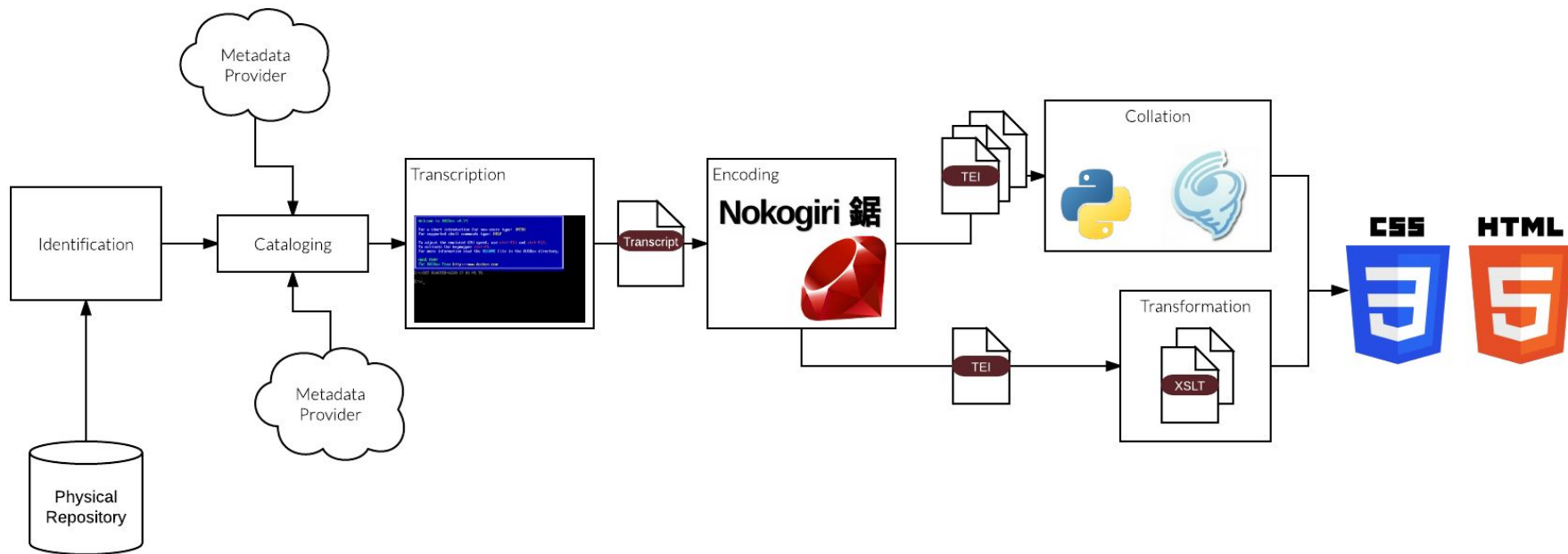
Contact Us

James Woolley (woolleyj@lafayette.edu)

Stephen Karian (karians@missouri.edu)

James R. Griffin III (griffinj@lafayette.edu)

Appendix: Workflow and System Architecture



Appendix: Collation Engine

- Solutions for collating variants of the transcribed texts were explored
 - Juxta
 - Supporting the integration of the Juxta API was given the highest priority
 - Given existing infrastructure and resource concerns there were limitations:
 - Preprocessing and postprocessing the TEI-XML Documents was necessary
 - Juxta itself required performance optimization for our environment
 - CollateX
 - An extremely viable solution
 - Mature (and maintained) Module for Python
 - Interoperability issues in supporting the features of the legacy interface
 - Concerns over whether preprocessing or postprocessing would be required
 - These concerns may not be warranted

Appendix: Collation Engine

- Prototyping a collation application in Python
 - The Tornado framework offered several advantages
 - Support for multiprocessing in collating larger sets of TEI-XML
 - Support for WebSockets (enabling asynchronous updates for a collation job)
 - Python Modules used to extend the features for the collation could be used
 - Natural Language Toolkit (supporting extensible tokenization)
 - NetworkX (supporting the building of stemmatic trees)
 - Integration with API's for XML databases could also be explored
 - eXistdb
 - Zorba
 - PostgreSQL