Making Choices About Your Data

Paige Morgan
Yvonne Lam
Welcome to DHSI 2018!

Thanks for joining the DHSI community!

In this booklet, you will find essential course materials prefaced by some useful information about getting settled initially at UVic, finding your way around, getting logged in to our network (after you’ve registered the day before our courses begin), and so on.

Given our community’s focus on things computational, it will be a surprise to no one that we might expect additional information online for some of the classes - your instructors will let you know - or that the most current version of all DHSI-related information may be found on our website at dhsi.org.

To access the DHSI wifi network, simply go into your wireless settings and connect to the “DHSI” network and enter the password “dhsi2018”.

And please don’t hesitate to be in touch with us at institut@uvic.ca or via Twitter at @AlyssaA_DHSI or @DHInstitute if we can be of any help ....
The 2018 schedule is just about ready! A very few things to confirm, add, etc, but this is the place to be to find out what is happening when / where ...

Psst: Some Suggested Outings

If you're here a day or two before we begin, or staying a day or two afterwards, here are a few ideas of things you might consider doing ....

▼ Suggested Outing 1, Botanical Beach (self-organised; car needed)
A self-guided visit to the wet, wild west coast tidal shelf (and historically-significant former research site) at Botanical Beach; we recommend departing early (around 8.00 am) to catch low tide for a better view of the wonderful undersea life! Consider bringing a packed lunch to nibble-on while looking at the crashing waves when there, and then have an afternoon drink enjoying the view from the deck of the Port Renfrew Hotel.

▼ Suggested Outing 2, Butchart Gardens (self-organised)
A shorter journey to the resplendently beautiful Butchart Gardens and, if you like, followed by (ahem) a few minutes at the nearby Church and State Winery, in the Saanich Peninsula. About an hour there by public bus from UVic, or 30 minutes by car.

▼ Suggested Outing 3, Saltspring Island (self-organised; a full day, car/bus + ferry combo)
Why not take a day to explore and celebrate the funky, laid back, Canadian gulf island lifestyle on Saltspring Island. Ferry departs regularly from the Schwartz Bay ferry terminal, which is about one hour by bus / 30 minutes by car from UVic. You may decide to stay on forever ....

▼ Suggested Outing 4, Paddling Victoria's Inner Harbour (self-organised)
A shorter time, seeing Victoria's beautiful city centre from the waterways that initially inspired its foundation. A great choice if the day is sunny and warm. Canoes, kayaks, and paddle boards are readily rented from Ocean River Adventures and conveniently launched from right behind the store. Very chill.

And more!
Self-organised High Tea at the Empress Hotel, scooter rentals, visit to the Royal BC Museum, darts at Christies Carriage House, a hangry breakfast at a local diner, whale watching, kayaking, brew pub sampling (at Spinnaker's, Swans, Moon Under Water, and beyond!), paddle-boarding, a tour of used bookstores, and more have also been suggested!

Sunday, 3 June 2018 [DHSI Registration + Suggested Outings]

9:00 to 4:00
▼ Early Class Meeting: 4. [Foundations] DH For Department Chairs and Deans (Hickman 120, Classroom)
Further details are available from instructors in mid May to those registered in the class. Registration materials will be available in the classroom.

3:00 to 5:00
DHSI Registration (MacLaurin Building, Room A100)
After registration, many will wander to Cadboro Bay and the pub at Smuggler's Cove OR the other direction to Shelbourne Plaza and Maude Hunter's Pub OR even into the city for a nice meal.

Monday, 4 June 2018

Your hosts for the week are Alyssa Arbuckle, Ray Siemens, and Dan Sondheim.

7:45 to 8:15
Last-minute Registration (MacLaurin Building, Room A100)

8:30 to 10:00
Welcome, Orientation, and Instructor Overview (MacLaurin A144)
Abstract: I've spent a lot of time in the last 12 months thinking about fascism, digital humanities, its long histories, and what it means to do DH work that centers social justice particularly in this global rise of late fascism. I will speak briefly about DH's history, including the medieval history related to Busa but how that connects to data systems that created the Holocaust and also participated in the Cold War nuclear military complex.

military complex.

history really connects to data systems that created the Holocaust and also participated in the Cold War nuclear military complex.

Institute Panel: Perspectives on DH (or, #myDHis …)
Chair: Alyssa Arbuckle (U Victoria)

Dorothy Kim (Vassar C): "#MyDHis Antifascist."

Lee Zickel (Case Western Reserve U): "Comfortably Trepid."
Abstract: #myDHis found outside the well-established, DH-friendly institutions, at an institution that is devoted predominantly to Medicine and Engineering. I, and with increasing frequency other DH practitioners and instructors, am not positioned in a DH Lab or Humanities Center, but in ITS. Part teacher, part technologist, part translator, I will briefly discuss my work supporting humanists and social scientists, particularly those who are new to or less comfortable with computational methodologies.

Margaret Konkol (Old Dominion U): "Prototyping Mina Loy’s Alphabet with a 3D Printer.”
Abstract: This talk discusses the interpretive and methodological implications of using 3D printing technologies to prototype the archival diagrams of a proposed but never constructed plastic segmental alphabet letter kit—a game designed by modernist poet Mina Loy for F.A.O Schwarz. Although intended as a toy for young children, the Alphabet that Builds Itself, as a work of “object typography” articulates a theory of language as kinetic, geometric, recombinant, and open to mutation. Alphabetic segments extend into the x, y, and z coordinates in exponential iterations and conjoin with magnets. Combining elements of contemporaneous typefaces like Futura and Gil Sans, which represented modernity’s functional ideals and democratic principles of simplicity, these recombinant letters represent, as this talk argues, Loy’s unpublished modernist poem, an articulation of Loy’s concept of language as a physical fact in which substance, not just form, is semantic.

Emily Murphy (U Victoria): "#MyDHis Edgy.”
Abstract: I will build upon—or, possibly, perform a misprision of—a tweet by Polina Vinogradova; "#myDHis messy, dusty, edgy, and radically inclusive!" Vinogradova evokes the mess and dust of the archives, the edges that connect nodes of a network, and the political impetus to think of cultural history and community together. I argue that these aspects of DH have a renewed importance as we head into a moment of feminist historiography.

Milena Radzikowska (Mt Royal C): “Release the Kraken: Story-Driven Prototyping for the Digital Humanities.”
Abstract: I have spent the last 15 years of my career designing text analysis tools for use by humanities scholars. In this brief presentation, I propose to share a concept-based approach to interface design for DH.

Emily Murphy (U Victoria): "#MyDHis Edgy.”
Abstract: I will build upon—or, possibly, perform a misprision of—a tweet by Polina Vinogradova; "#myDHis messy, dusty, edgy, and radically inclusive!" Vinogradova evokes the mess and dust of the archives, the edges that connect nodes of a network, and the political impetus to think of cultural history and community together. I argue that these aspects of DH have a renewed importance as we head into a moment of feminist historiography.

Margaret Konkol (Old Dominion U): "Prototyping Mina Loy’s Alphabet with a 3D Printer.”
Abstract: This talk discusses the interpretive and methodological implications of using 3D printing technologies to prototype the archival diagrams of a proposed but never constructed plastic segmental alphabet letter kit—a game designed by modernist poet Mina Loy for F.A.O Schwarz. Although intended as a toy for young children, the Alphabet that Builds Itself, as a work of “object typography” articulates a theory of language as kinetic, geometric, recombinant, and open to mutation. Alphabetic segments extend into the x, y, and z coordinates in exponential iterations and conjoin with magnets. Combining elements of contemporaneous typefaces like Futura and Gil Sans, which represented modernity’s functional ideals and democratic principles of simplicity, these recombinant letters represent, as this talk argues, Loy’s unpublished modernist poem, an articulation of Loy’s concept of language as a physical fact in which substance, not just form, is semantic.

Lee Zickel (Case Western Reserve U): "Comfortably Trepid.”
Abstract: #myDHis found outside the well-established, DH-friendly institutions, at an institution that is devoted predominantly to Medicine and Engineering. I, and with increasing frequency other DH practitioners and instructors, am not positioned in a DH Lab or Humanities Center, but in ITS. Part teacher, part technologist, part translator, I will briefly discuss my work supporting humanists and social scientists, particularly those who are new to or less comfortable with computational methodologies.

Dorothy Kim (Vassar C): "#MyDHis Antifascist.”
Abstract: I've spent a lot of time in the last 12 months thinking about fascism, digital humanities, its long histories, and what it means to do DH work that centers social justice particularly in this global rise of late fascism. I will speak briefly about DH's history, including the medieval history related to Busa but how that history really connects to data systems that created the Holocaust and also participated in the Cold War nuclear military complex.
Randa El Khatib (U Victoria): "Learning from the Iterative Process."
Abstract: #MyDHis Iterative. In addition to the improvements that come with iterative projects, the iterative process itself is a fruitful area for scholarly inquiry. Within this iterative context, the various teams that I work with and I have been reflecting on and rethinking central DH practices, such as what it means to collaborate, prototype, remix, and implement DH values in our work. In this talk, I will present the various lessons learnt along the way.

Sarah Melton (Boston C): "#MyDHis...People."
Abstract: Taking seriously Miriam Posner’s exhortation to “commit to DH people, not DH projects,” I invite us to reflect on how people are the core of DH. In this brief talk, I will explore the intersections between DH, labor, and infrastructure.

Tuesday, 5 June 2018

5:00 to 6:00 Opening Reception (University Club)
We are grateful to Gale Cengage for its sponsorship.

9:00 to Noon Classes in Session

12:15 to 1:15 Lunch break / Unconference
"Mystery" Lunches
DHSI Lunchtime Workshop Session (click for workshop details and free registration for DHSI participants)
73. Introduction to ORCID (Digital Scholarship Commons, Classroom).

1:30 to 4:00 Classes in Session

4:15 to 5:15 DHSI Colloquium Lightning Talk Session 1 (MacLaurin A144)
Chair: James O’Sullivan
- New Modes of DH and Archival Skills Acquisition in a Graduate Public History Course. Paulina Rousseau (Ryerson U)
- Walking a Transect: Exploring a Soundscape. John Barber (Washington State U)
- Centering the Edge Case: Designing Services for Humanities Data Research. Grace Afsari-Mamagani (New York U)
- Orwellian Vocabulary and the 21st-Century Politics. Ilgin Kizilgunesler (U Manitoba)
- Making Open Data from a Gray Archive. Sara Palmer (Emory U)

6:00 to 8:00 DHSI Newcomer’s Beer-B-Q (Felicitas, Student Union Building)

Wednesday, 6 June 2018

9:00 to Noon Classes in Session

Lunch break / Unconference
"Mystery" Lunches
Brown Bag Lecture: Alexandra Branzan Albu (U Victoria): “Visual Recognition of Symbolic and Natural Patterns” (Digital Scholarship Commons, 3rd Floor McPherson Library)

Abstract: Image-based object recognition is a visual pattern recognition problem; one may characterize visual patterns as either symbolic or natural. Symbolic patterns evolved for human communication; they include but are not limited to text, forms, tables, graphics, engineering drawings etc. Symbolic patterns vary widely in terms of size, style, language, alphabet and fonts; however, literate humans can easily compensate for this variability and instantly recognize most symbolic patterns. On the other hand, natural patterns characterize images of physical structures; they often lack the intrinsic discriminability and structure of symbolic patterns, and vary widely in terms of pose, perspective, and lighting.

This lecture will explore similarities and differences in approaches designed for recognizing visual and symbolic patterns, and will address the following questions via examples.
- What are the distinctive characteristics of natural patterns? What dimensions of variability can we infer?
- What are the distinctive characteristics of symbolic patterns? What dimensions of variability can we infer?

Alexandra Branzan Albu is an Associate Professor with the Department of Electrical and Computer Engineering and cross-listed with Computer Science. Her research interests are related to image analysis, computer vision, and visual computing. She is actively pursuing outreach activities dedicated to increasing the women's presence in electrical engineering and computer science.
<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Location</th>
<th>Notes</th>
</tr>
</thead>
</table>
| 4:15 to 5:15 | DHSI Colloquium Lightning Talk Session 2 (MacLaurin A144)                              | Chair: James O'Sullivan           | • Defining a Taxonomy of Abandonment for Online Digital Humanities Projects. Luis Meneses (Electronic Textual Cultures Lab, U Victoria) and Jonathan Martin (King's College London)  
• The Stories We Tell: Representing Gay and Lesbian History through Digital Technologies in the LGLC Project. Nadine Boulay (Simon Fraser University) and Ewan Matthews (Ryerson U)  
• Italian Paleography in the Digital Domain. Isabella Magni (Newberry Library)  
• Digital Humanities, A Question of Ethics. Negar Basiri (Louisiana State U)  
• Writing Poetry in High School. Guadalupe Echegoyen (National Autonomous U Mexico) |
| 6:00 to 7:00 | “Half Way There!” [An Informal, Self-Organized Birds of a Feather Get-Together] (Feligitas, Student Union Building)  
Bring your DHSI nametag and enjoy your first tipple on us! |                                    |                                            |
| 9:00 to Noon | Classes in Session                                                                      |                                   |                                            |
| 12:15 to 1:15| Lunch break / Unconference  
“Mystery” Lunches                                                                           | UVIC Library/ETCL lunchtime talk: “A Humanities Application of 3D printing and Machine Translation in the ChessBard and Loss Sets” by Dr. Aaron Tucker Digital Scholarship Commons, 3rd floor, Mearns Centre for Learning / McPherson Library  
Bring your lunch and come on up! (Instructor lunch meeting) |                                            |
| 1:30 to 4:00 | Classes in Session                                                                      |                                   |                                            |
| 4:15 to 5:15 | DHSI Colloquium Lightning Talk Session 3 (MacLaurin A144)                              | Chair: James O'Sullivan           | • Documenting Deportation: A Collaborative Digital Collection. Paulina Rousseau (Ryerson U)  
• Unleashing the Power of Texts as Networks: Visualizing the Scholastic Commentaries and Texts Archive. Jeffrey Witt (Loyola U Maryland) and Drew Winget (Stanford U)  
• #haunteDH: Punching holes in the International Busa Machine Narrative. Arun Jacob (McMaster U)  
• Text in World: Computational Analysis of Trauma in Genocide Narratives. Nanditha Narayanamoorthy (U York) and Krish Perumal (U Toronto) |
| 7:30 to 9:30 | (Groovy?) Movie Night (MacLaurin A144)                                                  |                                   |                                            |
| 9:00 to Noon | DHSI Classes in Session                                                                 |                                   |                                            |
| 12:15 to 1:15| DHSI Lunch Reception / Course E-Exhibits (MacLaurin A100)                               |                                   |                                            |
| 1:00 to 2:00 | DLFxDHSI Registration (MacLaurin A100)                                                  |                                   |                                            |
| 1:30 to 1:50 | [DHSI] Remarks, A Week in Review (MacLaurin A144)                                      |                                   |                                            |
| 2:00 to 3:00 | Joint Institute Lecture (DHSI and DLFxDHSI): Bethany Nowviskie (CLIR DLF and U Virginia): “Reconstitute the World: Machine-reading Archives of Mass Extinction”  
Chair: Lisa Goddard (U Victoria)  
(MacLaurin A144)  
Abstract: The basic constitution of our digital collections becomes vastly more important in the face of two understandings: first, that archives of modernity are archives of the sixth great mass extinction of life on our planet; and next, that we no longer steward cultural heritage for human readers alone. In the same way that people are shaped by what we read, hear, and see, the machine readers that follow us into and perhaps beyond the Anthropocene have begun to learn from “unsupervised” encounters with our digital libraries. What will we preserve for the living generations and artificial intelligences that will come? What do we neglect, or even choose to extinguish? And from an elegiac archive, a library of endings, can we create forward-looking, speculative collections—collections from which to deep-dream new futures? The most extra/ordinary power we possess is the power to make poetry from records of the past. Could it be called on, one day, to reconstitute the world? | (MacLaurin A144) |                                            |

Thursday, 7 June 2018

Friday, 8 June 2018 [DHSI; DLFxDHSI Opening]
Saturday, 9 June 2018 [DLFxDHSI + DHSI Conference and Colloquium]

8:30 to 9:00 DLFxDHSI Registration (MacLaurin A100)

9:00 to 5:30 DLFxDHSI UnConference Sessions

➤ DHSI All Day Workshop Session (click for workshop details and free registration for DHSI participants)

9:00 to 4:00

• 53. Building Your Academic Digital Identity (MacLaurin D105, Classroom)

• DHSI Colloquium Day Conference (MacLaurin A144)

Welcome

People I: Documenting Online Lives. Chair: Molly Nebiolo (University of New York)

- Examining Gendered Harassment Online and in Silicon Valley. Andrea Flores (Ulta College)
- This is Just to Say I Have <X> the <Y> in your <Z>: Modernist Memes in an Era of Public Apology. Shawna Ross (Texas A&M University)

Break

People II: Documenting Lives Online. Chair: Dheepa Sundaram (College of Wooster)

- Youtube Yoga and Ritual on Demand: The Virtual Economics of Hindu Soteriology. Dheepa Sundaram (College of Wooster)
- The Resemblage Project: Creativity and Digital Health Humanities in Canada. Andrea Charise (University of Toronto) and Stefan Krecsy (University of Toronto)

Lunch

Projects I: Building and Analyzing. Chair: Yannis Rammos (New York University)

- Building the ARTECHNE Database: New directions in Digital Art History. Marieke Hendriksen (Old Dominion University)
- The Ineffective Inquisition: The Holy Office’s Sphere of Influence in Early Modern New Spain. Kira Homo (Pennsylvania State University)

Break

Projects II: Mapping and Visualizing. Chair: Innocent Opara (Qumet Institute)

- Mapping Sarah Sophia Bank’s Numismatic Collection. Erica Hayes (North Carolina State University) and Kacie Wills (University of California, Riverside)
- Text Mining and Visualizing 18th Century American Correspondence. Ashley Sanders Garcia (University of California, Los Angeles)

Break

Practices: Digital Scholarship on Campus and in the Classroom. Chair: Alyssa Arhuckle (University of Victoria)
Concluding Remarks

**8:30 to 9:00** Symposium on Indigenous New Media Registration (MacLaurin A100)

**9:00 to 5:00** DHSI Registration (MacLaurin A100)

**9:00 to 4:00** SINM Sessions
- 63. Symposium on Indigenous New Media: Reading Group (Hickman 105, Classroom)
- 72. Symposium on Indigenous New Media: Indigitization (Hickman 120, Classroom)

**9:00 to 4:00** DHSI All Day Workshop Sessions (click for workshop details and free registration for DHSI participants)
- 53. Building Your Academic Digital Identity (MacLaurin D105, Classroom)
- 54. An Introduction to the Archaeology of 1980s Computing (MacLaurin D114, Classroom)

**9:00 to Noon** DHSI AM Workshop Sessions (click for workshop details and free registration for DHSI participants)
- 55. Regular Expressions (MacLaurin D111, Classroom)
- 56. 3D Visualization for the Humanities (MacLaurin D010, Classroom)
- 58. DH Fieldwork Methods (MacLaurin D016, Classroom)
- 60. Pedagogy of the Digitally Oppressed: Inculcating De-/Anti-/Post-Colonial Digital Humanities (MacLaurin D107, Classroom)
- 61. Introduction to #GraphPoem. Digital Tools for Poetry Computational Analysis and Graph Theory Apps in Poetry (MacLaurin D101, Classroom)
- 62. Creating a CV for Digital Humanities Makers (MacLaurin D115, Classroom)

**1:00 to 4:00** DHSI PM Workshop Sessions (click for workshop details and free registration for DHSI participants)
- 64. Agent-Based Modelling in the Humanities (MacLaurin D111, Classroom)
- 65. Unleash Linux on macOS (MacLaurin D010, Classroom)
- 66. DHSI Knits: History of Textiles and Technology (MacLaurin D016, Classroom)
- 67. Crowdsourcing as a Tool for Research and Public Engagement (MacLaurin D109, Classroom)
- 69. Web Annotation as Critical Humanities Practice (MacLaurin D103, Classroom)
- 70. Dynamic Ontologies for the Humanities (MacLaurin D107, Classroom)
- 71. Social Media Research in the Humanities (MacLaurin D109, Classroom)

**4:10 to 5:00** Joint Institute Lecture (DHSI and SINM):
David Gaertner (U British Columbia): "A Landless Territory?: CyberPowWow and the Politics of Indigenous New Media."
Chair: Deanna Reder (Simon Fraser U) (MacLaurin A144)

Abstract: Following the 1997 launch of Skawennati’s (Mohawk) CyberPowWow, digital space has become a vital new territory for the resurgence of Indigenous storytelling and cultural practice: "We have signed a new treaty,” Cree artist Archer Pechawis wrote of this period, “and it is good. We have the right to hunt, fish, dance and make art at www.CyberPowWow.net, .org and .com for as long as the grass grows and the rivers flow.” This talk will critically explore the theoretical, cultural, political-economic, and gendered dynamics underlying the histories and futures of Indigenous new media. Particular attention will be given in examining the ways in which new media and digital storytelling connect to and support key issues in the field of Indigenous studies, such as sovereignty, self-determination, decolonization, and land rights.

After the day, many will wander to Cadboro Bay and the pub at Smuggler's Cove OR the other direction to Shelbourne Plaza and Maude Hunter's Pub OR even into the city for a bite to eat.

**Sunday, 10 June 2018 [SINM + DHSI Registration, Workshops]**

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30 to 9:00</td>
<td>Symposium on Indigenous New Media Registration (MacLaurin A100)</td>
</tr>
<tr>
<td>9:00 to 5:00</td>
<td>DHSI Registration (MacLaurin A100)</td>
</tr>
<tr>
<td>9:00 to 4:00</td>
<td>SINM Sessions</td>
</tr>
<tr>
<td>9:00 to 4:00</td>
<td>DHSI All Day Workshop Sessions (click for workshop details and free registration for DHSI participants)</td>
</tr>
<tr>
<td>9:00 to Noon</td>
<td>DHSI AM Workshop Sessions (click for workshop details and free registration for DHSI participants)</td>
</tr>
<tr>
<td>1:00 to 4:00</td>
<td>DHSI PM Workshop Sessions (click for workshop details and free registration for DHSI participants)</td>
</tr>
</tbody>
</table>

**Monday, 11 June 2018 [DHSI + SINM]**
<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:45 to 8:15</td>
<td>DHSI Last-minute Registration</td>
<td>MacLaurin A100</td>
</tr>
<tr>
<td>8:30 to 10:00</td>
<td>DHSI Welcome, Orientation, and Instructor Overview</td>
<td>MacLaurin A144</td>
</tr>
<tr>
<td>9:00 to 4:00</td>
<td>SINM Sessions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ DHSI Classes in Session (click for details and locations)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 27. [Foundations] Understanding The Predigital Book: Technology and Texts (McPherson Library A003, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 28. [Foundations] Developing a Digital Project (With Omeka) (Clearihue D132, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 29. [Foundations] Models for DH at Liberal Arts Colleges (&amp; 4 yr Institutions) (MacLaurin D109, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 32. Stylometry with R: Computer-Assisted Analysis of Literary Texts (Clearihue A102, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 33. Digital Storytelling (MacLaurin D111, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 34. Text Mapping as Modelling (Clearihue D131, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 35. Geographical Information Systems in the Digital Humanities (Clearihue A105, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 36. Open Access and Open Social Scholarship (MacLaurin D114, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 37. Introduction to Machine Learning in the Digital Humanities (Cornett A229, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 38. Queer Digital Humanities: Intersections, Interrogations, Iterations (MacLaurin D110, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 41. Using Fedora Commons / Islandora (Human and Social Development A160, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 42. Documenting Born Digital Creative and Scholarly Works for Access and Preservation (MacLaurin D115, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 43. Games for Digital Humanists (MacLaurin D016, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 44. XPath for Document Archeology and Project Management (Cornett A128, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 46. Surveillance and the Digital Humanities (MacLaurin D103, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 47. Text Analysis with Python and the Natural Language ToolKit (Clearihue A103, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 48. Information Security for Digital Researchers (Clearihue D130, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 49. Wrangling Big Data for DH (Human and Social Development A150, Lab)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 50. Accessibility &amp; Digital Environments (MacLaurin D101, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 51. Critical Pedagogy and Digital Praxis in the Humanities (MacLaurin D105, Classroom)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 52. Drupal for Digital Humanities Projects (MacLaurin D107, Classroom)</td>
<td></td>
</tr>
<tr>
<td>10:15 to Noon</td>
<td>Lunch break / Unconference Coordination Session</td>
<td>MacLaurin A144</td>
</tr>
<tr>
<td></td>
<td>(Grab a sandwich and come on down!)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DHSI Undergraduate Meet-up, Brown-Bag (details via email)</td>
<td></td>
</tr>
<tr>
<td>12:15 to 1:15</td>
<td>Lunch break / Unconference Coordination Session</td>
<td>MacLaurin A144</td>
</tr>
<tr>
<td></td>
<td>(Grab a sandwich and come on down!)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DHSI Undergraduate Meet-up, Brown-Bag (details via email)</td>
<td></td>
</tr>
<tr>
<td>1:30 to 4:00</td>
<td>DHSI Classes in Session</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ Joint Institute Lecture (DHSI and SINM):</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jordan Abel (Simon Fraser U): “Indigeneity, Conceptualism, and the Borders of DH.” Chair: Michelle Brown (U Hawaii) (MacLaurin A144)</td>
<td></td>
</tr>
<tr>
<td>4:10 to 5:00</td>
<td>Joint Reception: DHSI and SINM</td>
<td>University Club</td>
</tr>
<tr>
<td>5:00 to 6:00</td>
<td>Joint Reception: DHSI and SINM</td>
<td>University Club</td>
</tr>
<tr>
<td>9:00 to Noon</td>
<td>Classes in Session</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lunch break / Unconference</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;Mystery&quot; Lunches</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ DHSI Lunchtime Workshop Session (click for workshop details and free registration for DHSI participants)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 73. Introduction to ORCID (Digital Scholarship Commons, Classroom)</td>
<td></td>
</tr>
</tbody>
</table>

**Tuesday, 12 June 2018**
Wednesday, 13 June 2018

9:00 to Noon  Classes in Session

12:15 to 1:15  Lunch break / Unconference
               "Mystery" Lunches

1:30 to 4:00  Classes in Session

4:15 to 5:15  DHSI Colloquium Lightning Talk Session 4 (MacLaurin A144)
              Chair: Lindsey Seatter
              - Mapping Indigenous and Chicana/o Environmental Imaginaries using GIS. Stevie Ruiz (California State U, Northridge), Quetzalli Enrique (California State U, Northridge), Enrique Ramirez (California State U, Northridge), and Tomas Figueroa (California State U, Northridge)
              - "But is it any good?": A quantitative approach to the popularity of digital fanfiction. Suzanne Black (U Edinburgh)
              - The American Prison Writing Archive (APWA). Doran Larson (Hamilton C), Janet Simons (Digital Humanities Initiative, Hamilton C), and William Rasenberger (Hamilton C)

6:00 to 8:00  DHSI Newcomer's Beer-B-Q (Felicitas, Student Union Building)

Thursday, 14 June 2018

9:00 to Noon  Classes in Session

12:15 to 1:15  Lunch break / Unconference
               "Mystery" Lunches

1:30 to 4:00  Classes in Session

4:15 to 5:15  DHSI Colloquium Lightning Talk Session 5 (MacLaurin A144)
              Chair: Lindsey Seatter
              - Faraway, so close: Has the political environment really changed in Ecuador?. Luis Meneses (Electronic Textual Cultures Lab, U Victoria)
              - Re-mixing Melville's Reading: Text Analysis of Marginalia with R and XSLT. Christopher Ohge (U London, School of Advanced Study) and Steven Olsen-Smith (Boise State U)
              - Developing Interactive and Open-Source OER: Inquiry-Based Music Theory. Evan Williamson (U Idaho)
              - Spatial Humanities and the Web of Everywhere. Ken Cooper (SUNY Geneseo)

6:00 to 7:00  "Half Way There (yet again)!" [An Informal, Self-Organized Birds of a Feather Get-Together] (Felicitas, Student Union Building)
              Bring your DHSI nametag and enjoy your first tipple on us!

Friday, 15 June 2018

9:00 to Noon  Classes in Session

12:15 to 1:15  Lunch Reception / Course E-Exhibits (MacLaurin A100)
1:30 to 2:30

Institute Lecture: William Bowen (U Toronto Scarborough): “Discovery, Collaboration and Dissemination: Lessons Learned and Plans for the Future” (MacLaurin A144)

Abstract: Much has changed and continues to change in digital humanities since the formal establishment of Iter in the Fall of 1997. However, the mandate of the not-for-profit partnership to support “the advancement of learning in the study and teaching of Middle Ages and Renaissance (400–1700) through the development and distribution of online resources” continues to have relevance. This presentation explores the striking challenges faced by Iter and presents our current thinking on the realization of this mandate for the future through a platform with a focus on facilitating the discovery of the academic resources necessary to our work; creating an environment for collaboration, sharing and developing projects; and on enabling the distribution and publication of our scholarship.

2:40 to 3:00

Awards and Bursaries Recognition
Closing, DHSI in Review (MacLaurin A144)

Contact info:
institut@uvic.ca  P: 250-472-5401  F: 250-472-5681
Schedule

Monday AM (shorter day because of opening session):

- Brief Introductions
- MEALS: why we use it to talk about data
- Goals for this workshop:
- Exercise: You Create the Data

Monday PM:

- Some vocab and principles:
  - Unstructured vs structured data
  - Four types of data models
    - hierarchical meta-markup languages
    - tabular
    - relational
    - non-relational
  - Source vs. method-oriented approaches to data
  - Controlled vocabularies and ontologies
- Introducing your datasets: why have you chosen them to work with?
- Reading:
  - Weigel, Moira “Silicon Valley’s Sixty-Year Love Affair With the Word Tool”, The New Yorker
  - Merry, Mark, “Designing Databases for Historical Research”, Sections A-C3.
  - Posner, Miriam, “Humanities Data: A Necessary Contradiction”
  - Digital Yoknapatawpha, “Instructions” (feel free to skim this: take time to read one or two sections more carefully, but don’t feel as though you need to have a grasp of the whole set of instructions.)

Tuesday AM:

- Reading:
  - Zyvagintseva, Lydia, “Organizing historical menus: a data curation experiment”
  - Muñoz, Trevor, “What IS on the menu?: More work with NYPL’s open data, Part One”
  - Muñoz, Trevor, “Refining the Problem: More work with NYPL’s open data, Part Two”
Understanding Platform Choice: Tool Introductions
Free work time

Tuesday PM:
- Reading:
  - Onuoha, Mimi, "Missing Datasets"
  - Thorp, Jer, “You Say Data, I Say System”
- Key concept: interoperability
  - Adopting aspects of existing datasets
  - Learning how and where to find those datasets
  - Documentation strategies

Wednesday AM:
- Cleaning and Enhancing Data with OpenRefine + Free work time

Wednesday PM:
- Reading:
  - Rawson, Katie, and Trevor Muñoz, “Against Cleaning”
- Free work time

Thursday AM:
- Reading:
  - D'Ignazio, Catherine, “What Would Feminist Data Visualization Look Like?”
  - Zepel, Tara, “Visualization as a Digital Humanities________?”
- Combining multiple tools
- Free work time

Thursday PM:
- Free work time

Friday AM:
- Class show & tell / show & tell prep
- Next steps: sharing and/or publishing your data

Additional (Optional) Included Readings


Sample Datasets

All sample datasets will be downloadable from a Google Drive Folder at https://bit.ly/2HzYbp6
In the written remarks that Mark Zuckerberg, the C.E.O. of Facebook, submitted in advance of his testimony on Capitol Hill this week, he used the word “tool” eleven times. “As Facebook has grown, people everywhere have gotten a powerful new tool to stay connected to the people they love, make their voices heard, and build communities and businesses,” Zuckerberg wrote. “We have a responsibility to not just build tools, but to make sure those tools are used for good.” Later, he added, “I don’t want anyone to use our tools to undermine democracy.” In his testimony before the Senate Judiciary
and Commerce Committees on Tuesday, Zuckerberg referred to “these tools,” “those tools,” “any tool,” “technical tools,” and—thirteen times—“A.I. tools.” On Wednesday, at a separate hearing of the House Energy and Commerce Committee, a congressman from Florida told Zuckerberg, “Work on those tools as soon as possible, please.”

What’s in a tool? The Oxford English Dictionary will tell you that the English word is more than a thousand years old and that, since the mid-sixteenth century, it has been used as the slur that we’re familiar with today. (Translating the Roman poet Juvenal, in 1687, the satirist Thomas Shadwell cut to the point: “What pleasure can the weak Old Doting Fool, / Expect from that infirm and Aged Tool?”) In Silicon Valley, according to Siva Vaidhyanathan, a professor at the University of Virginia whose book about Facebook, “Antisocial Media,” is due out in September, “Tools are technologies that generate other technologies.” When I asked an engineer friend who builds “developer tools” for his definition, he noted that a tool is distinct from a product, since a product is “experienced rather than used.” The iTunes Store, he said, is a product: “there are lots of songs you can download, but it’s just a static list.” A Web browser, by contrast, is a tool, because “the last mile of its use is underspecified.”

Yesterday was not Zuckerberg’s first time being called in and interrogated about a Web site that he created. In the fall of 2003, when he was a sophomore at Harvard, a disciplinary body called the Ad Board summoned him to answer questions about Facemash, the Facebook precursor that he had just released. Using I.D. photos of female undergraduates scraped from the university’s online directories, Facemash presented users with pairs of women and asked them to rank who was “hotter.” (“Were we let in for our looks? No,” the site proclaimed. “Will we be judged on them? Yes.”) By 10 p.m. on the day Facemash launched, some four hundred and fifty visitors had cast at least twenty-two thousand votes. Several student groups, including Fuerza Latina and the Harvard Association of Black Women, led an outcry. But Zuckerberg insisted to the Ad Board that he had not intended to “insult” anyone. As the student newspaper, the Crimson, reported, “The programming and algorithms that made the site function were Zuckerberg’s primary interest in creating it.” The point of Facemash was to make a tool. The fact that it got sharpened on the faces of fellow-students was incidental.
This was the first iteration of an excuse that Zuckerberg and Facebook have offered many times since, which was distilled in a recently leaked internal memo from 2016. In the memo, titled “The Ugly,” Andrew Bosworth, a vice-president at the company, wrote:

We connect people.

That can be good if they make it positive. Maybe someone finds love. Maybe it even saves the life of someone on the brink of suicide.

So we connect more people.

That can be bad if they make it negative. Maybe it costs a life by exposing someone to bullies. Maybe someone dies in a terrorist attack coordinated on our tools.

And still we connect people.

The ugly truth is that we believe in connecting people so deeply that anything that allows us to connect more people more often is de facto good.

Zuckerberg criticized Bosworth’s memo when it leaked, and he criticized it again on Tuesday. (“I disagreed with it at the time that he wrote it,” Zuckerberg told Senator Lindsey Graham. “The vast majority of people internally did, too.”) But the core idea—that tools can be used well or badly, and that the people who make them are not responsible for how others use or misuse them—resonates with Zuckerberg’s frequent refrain that Facebook is a “neutral platform.” Vaidhyanathan said he doubted that the use of the word was deliberate. “It’s probably just how they’re talking around the office,” he told me. Still, it implies a set of ideas and ideals. “A tool is not a weapon,” Vaidhyanathan pointed out. “You would not usually call an AK-47 a tool.”
The exaltation of tools has a long history in the Bay Area, going back to the late nineteen-sixties, when hippie counterculture intersected with early experiments in personal computing. In particular, the word got its cachet from the “Whole Earth Catalog,” a compendium of product reviews for commune dwellers that appeared several times a year, starting in 1968, and then sporadically after 1972. Its slogan: “Access to tools.” The publisher of the “Catalog,” Stewart Brand—a Stanford-trained biologist turned hippie visionary and entrepreneur—would later call it “the first instance of desktop publishing.” Steve Jobs, in his 2005 commencement address at Stanford, described it as “one of the bibles of my generation.” The “Catalog,” Jobs said, was “Google in paperback form, thirty-five years before Google came along. It was idealistic, and overflowing with neat tools and notions.” Jobs’s biographer, Walter Isaacson, quotes Brand as saying that the Apple co-founder was a kindred spirit; in designing products, Jobs “got the notion of tools for human use.” With the rise of personal computing, the term “tools” migrated from communes to software. The generation of tech leaders who grew up taking P.C.s and the World Wide Web for granted nevertheless inherited an admiration for Brand. In 2016, for instance, Facebook’s head of product, Chris Cox, joined him onstage at the Aspen Ideas Festival to give a talk titled “Connecting the Next Billion.”

Tool talk encodes an entire attitude to politics—namely, a rejection of politics in favor of tinkering. In the sixties, Brand and the “Whole Earth Catalog” presented tools as an alternative to activism. Unlike his contemporaries in the antiwar, civil-rights, and women’s movements, Brand was not interested in gender, race, class, or imperialism. The transformations that he sought were personal, not political. In defining the purpose of the “Catalog,” he wrote, “a realm of intimate, personal power is developing—power of the individual to conduct his own education, find his own inspiration, shape his own environment, and share his adventure with whoever is interested.” Like Zuckerberg, Brand saw tools as a neutral means to engage any and every user. “Whole Earth eschewed politics and pushed grassroots direct power—tools and skills,” he later wrote. If people got good enough tools to build the communities they wanted, politics would take care of itself.

This idea became highly influential in the nineties, as the Stanford historian Fred Turner demonstrates in his book “From Counterculture to Cyberculture.” Through Wired magazine, which was founded by Brand’s collaborator Kevin Kelly, the message
reached not just Silicon Valley but also Washington. The idea that tools were preferable to politics found a ready audience in a decade of deregulation. The sense that the Web was somehow above or beyond politics justified laws that privatized Internet infrastructure and exempted sites from the kinds of oversight that governed traditional publishers. In other words, Brand’s philosophy helped create the climate in which Facebook, Google, and Twitter could become the vast monopolies that they are today—a climate in which dubious political ads on these platforms, and their casual attitudes toward sharing user data, could pass mostly unnoticed. As Turner put it in a recent interview with *Logic* magazine (of which I am a co-founder), Brand and *Wired* persuaded lawmakers that Silicon Valley was the home of the future. “Why regulate the future?” Turner asked. “Who wants to do that?”

For decades, the American public has largely embraced this optimistic, laissez-faire ideology. But a tech backlash has been building steam since the 2016 Presidential election, and especially since the revelation that Cambridge Analytica, a political-consulting firm connected with Donald Trump, surreptitiously harvested the data of eighty-seven million Facebook users. On Tuesday, senators directly challenged the idea that Facebook could ever be a “neutral platform.” It remains to be seen whether American politicians and the American public will accept Zuckerberg’s can-do talk of technical solutions—or whether they will dismiss him as a tool himself.

*Moira Weigel is a junior fellow at the Harvard Society of Fellows and a founding editor of *Logic*, a magazine about technology.*  
*Read more »*
Table of contents

Contents

A. Introduction
   A1. Database design concepts

B. Sources, information and data
   B1. Introduction
   B2. Information and data

C. Fundamentals of database design
   C1. Introduction
   C2. The Purpose of the database
   C3. Conceptual models of database design
   C4. Database ‘layers’
   C5. Database definitions – tables, fields, records, values, rules and datatypes
   C6. Conclusion

D. Relationships
   D1. Introduction
   D2. Functions of relationships
   D3. Types of relationship
   D4. Primary and Foreign Keys

E. Entity relationship modelling
   E1. Introduction
   E2. Entity Relationship Modelling step-by-step
   E3. Conclusion

F. Problems facing the historian
   F1. Introduction
   F2. Problematic information
   F3. Standardisation, classification and coding

G. Conclusion
Contents

By Mark Merry

A. Introduction
B. Sources, information and data
C. Fundamentals of database design
D. Relationships
E. Entity relationship modelling
F. Problems facing the historian
G. Conclusion
A. Introduction

A1. Database Design Concepts

The purpose of this Historical Research Handbook is to provide an introduction to designing databases for use in historical research. It will provide an overview of important concepts – both historical in nature and in terms of databases – that the historian will need to consider before embarking upon designing a database, and it will provide a number of starting points for overcoming certain design problems that specifically affect historians when they come to wrestle their sources into a database. This Handbook does not cover the actual construction of databases for historical research in any practical manner.
A1. Database design concepts

After working through this Handbook it is hoped that you will have a good understanding of the complex relationship between historical sources, information and data, and will be aware of the translation processes that are required when moving from one to the others. The informational contents of historical sources need to be converted – often in multiple ways – before they can be used as data, and a number of methodological decisions will need to be taken as this is done. Unlike the more mechanical aspects of using databases in historical research, such as building tables, linking records or running aggregate analyses, this translation process is not only difficult to learn other than through experience, it is also likely to be a substantially different process for every historian doing it. Each historian has different materials, different projects and different research aims, and so the databases they build will (or should) address these in the way that best fits their specific purpose. This ‘modelling’ of historical data is a difficult process, but happily the difficulties that arise are those that by and large are faced by historians in their everyday, non-database, work, meaning that you will be well equipped to deal with them. The modelling of data is also possibly the most interesting and enjoyable aspect of using databases in historical research, although perhaps this is only a relative response to the long hours of data entry that follow the initial design of the database!

This Handbook does not require the use of database software, although it will show the occasional screenshot of a database for the purposes of illustration. Instead it will spend most of its sections on discussing sources and research questions, and how these need to be recast when interacting with a database with its strictures and rules. The exercises that are offered will not have right or wrong ‘answers’, just as there is no right or wrong way to design a database of historical sources. Or rather, it may be more accurate to suggest that while there is no right way to design a database, there are a number of (if not wrong exactly) unhelpful ways to design a database, and this Handbook will focus as much on the latter as on the former.
B. Sources, information and data

B1. Introduction

B2. Information and Data
B1. Introduction

In this section we are going to address some of the issues that historians face when it comes to thinking about building and using a database for their research. Quite what ‘using a database for their research’ actually means is a subject that we will return to in Section C of this Handbook, as it is a subject that encompasses a range of issues which are likely to impact upon the design of a historical database. Essentially what this section will focus on is the difference between ‘information’ and ‘data’ – the former being what sources provide, the latter being what databases need – and it will begin the process of considering how to move from one to the other.

Unfortunately, the historian is faced with particular kinds of problem when it comes to converting sources into a useful database resource, problems which are not shared by most other database users. This (as we shall see) boils down to two separate inescapable realities of historical research:

- The historian often does not know precisely what kinds of analyses they want to conduct when starting out on their research
- The extent and scope of the information contained within the historian’s particular sources cannot usually be anticipated fully

In other words the sheer unpredictability of many historical research projects, the various tangents and new lines of inquiry that open up as soon as you get to grips with the sources, as well as the constant promise of unearthing a type of information that you were not expecting, make designing databases a difficult proposition for historians. Indeed, in many ways, these two factors provide conditions which are entirely contrary to the environment required by the structures and functions of a database. The difficulty for the historian is that what is required is to take information that is informal and unstructured, translate it, and make it fit into a rigidly formal and structured medium. Reconciling the two – the milieu of the historian and the rules of databases – is the principal aim of this Handbook.

Much of what is discussed here is about good practice and ensuring that the most common and critical mistakes are avoided at the most important stage of database creation. Errors at this juncture will have an effect on how useful the database will be: they will make data entry more laborious and more difficult; and more seriously, they will have a significant impact upon the database’s ability to retrieve data for analysis. It is very important therefore to design the database as ‘correctly’ as possible, initially, to minimise the need for retrospective restructuring further down the line (although some of this will inevitably be necessary).
B2. Information and data

The tricky part of the process of using databases in historical research lies in the ‘shape’ of the information that is found in our sources. Databases have very strict rules about what type of information goes where, how it is represented and what can be done with it (see Section C), and if the information from our sources can be made to obey these rules then it has become data. Of course the problem facing historians is that information can take many forms in our sources, even if the research is only considering a single type of source and that source is a relatively simple one. Sources that are irregular in ‘shape’, such as textual sources with long narrative accounts written in paragraphs and chapters and so on, or databases of image/sound/video collections, are particularly problematic when it comes to converting their information into data; but the problem will also arise in the more structured sources (such as census listings or taxation assessments), which are never quite as simple as they might appear.

The concept of ‘shape’ here is one that is fundamental to the understanding of how databases work and the efforts needed to enter our sources into them. One of the database rules alluded to above is that all data in the database sit in tables, regardless of what kind of data they are. This means that information taken from our sources will need to fit into a tabular structure – that is, arranged by rows and columns – by the time it has been entered into the database. Often this is not the ‘shape’ the information is in when we open the pages of our sources, and usually we will have to mould it into a more compliant shape. As we shall see, this will cause us to accept something of a compromise between maintaining the full richness and integrity of our sources’ information on the one hand, and maximising the analytical potential of the data we create on the other.

Information from our sources is what we are interested in. It is what we will use to perform our historical analyses, and it is the raw material of our research. Away from the database, when looking at our sources as a methodological necessity we extract information from them and record that information as notes (sometimes as transcripts) in a variety of forms. The recording of information in this way allows us access to what we need without having to consult the original source in the future, but the form of our notes also allows us to accommodate the vagaries in the types of information that we can obtain from the source. In making notes we assimilate the variations in the type and scope of the information being recorded without concern for the shape of that information, something that is no longer possible in a database environment.

For example, image B2i depicts an interesting historical source, eminently useful for researching a variety of social, economic, cultural or political subjects in the context of mid-nineteenth century Chicago. The text of this pamphlet provides the historian with the bulk of the source’s information - information about places, dates, themes and events and so on – but from the point of view of database design it is important to note that not all of the information is contained in the source’s text. It is important to identify these non-textual types of information (such as page dimensions, layout, font types and sizes, language, archival stamps, colours used etc.) because if they are important to your research then they will need to be accommodated within the database design, and in some cases this will involve extra conversion processes. Descriptions of the source can be useful information every bit as much as what the source actually says.

When considering this pamphlet as a candidate for inclusion into a database, the most obvious aspect of this particular source is that it does not look much like a table. It is not ‘rectangular’ in terms of its shape - the text is not organised into columns and rows. This makes it difficult to ascertain the scope of the information (what there is information about) without actually reading the whole source, in the same way that you might be able to with a source arranged by rows and columns in a database.
Immediately therefore it becomes apparent that if we wanted to include this information in our database, we would need to think carefully about how to enter the information we want into the tabular structure required. How would it be possible to reorder the information into columns and rows – what would our columns be, how could the information be divided into instances of something (rows)? Our sources, whilst they may be wonderfully useful things, are not often actually suited for use in databases.

On the other hand there are sources which are more promising at first glance in terms of their suitability for inclusion in a database. Take for example the returns of the census enumerators (such as that for the 1850 US Census, image B2ii), a source which is as ‘rectangular’ in shape as it is possible to be. Here the information is conveniently arranged into columns and rows – each columns pertains to one particular type of information (name, age, occupation and so on), and each row corresponds to information about a single individual. This is a source which will ‘fit’ into the database structure without the need for too much conversion, as its inherent shape approximates that required by the database quite closely.
However, it is worth noticing that even here the translation process between source information and database data will not necessarily be an entirely problem-free one. Whilst the bulk of the information is contained within the tabular structure of the source, not all of it is. The information at the top of the page for example, vital information about the place and date of the listing, as well as the identity of the enumerator, is not contained within the table of the individual returns. In the database of this page, this information would need to be accommodated within a table somewhere, giving us some thinking to do about how this should be managed. Similarly, there are a number of pieces of information which might be useful to our research that do not exist in the table of individual listings: the arrow pointing to the Lincoln household, for example, or the various ticks and crosses, emendations and marginalia, some of which are not original to the source but which still constitute information, might be desirable for inclusion in the database. As we shall see in Section C not all the information from a source need necessarily be
included in the database and significant decisions about this will need to be made, but the information that is required, no matter what its shape or where it is located in the source, will need to be appropriately converted before it can be used in the database.

Manuscript Exercise

The need to understand the differences between the shape that information takes in our sources and the shape that data has to adopt within databases, is something that this Handbook will return to repeatedly, from a variety of angles. Squeezing information into the right shape for use in a database is not the only form of conversion that is required, however, as we shall see in Section F, but it is the most fundamental stage of the process, and is the most important step in the design stages, as we shall see in Section E.


C. Fundamentals of database design

C1. Introduction

C2. The Purpose of the Database

C3. Conceptual Models of Database Design

C4. Database 'Layers'

C5. Database definitions - tables, fields, records, values, rules and datatypes

C6. Conclusion
C1. Introduction

This section of the Handbook provides an introduction to the basic concepts that underpin the design and use of databases. It will focus on aspects of the database which will need to be considered at the beginning of the database design process, alongside the identification of information in the sources (see Sections B and E). The ideas and approaches discussed here are independent of the sources being employed by the historian, and as such will apply to every database used in a historical research project (and indeed to most databases in general). This section will not address technology or software, but rather it will examine conceptual approaches to designing database in the abstract, and specifically how these concepts will affect databases to be created and used by historians.
C2. The Purpose of the database

As we shall see in Section E, the very first step in the formal process for designing a database is to decide what purpose(s) the database is to serve. This is something that is perhaps not as obvious or as straightforward as one might expect, given that databases in the abstract can indeed serve one or more of a number of different kinds of function. In essence, however, there are three types of function that the historian is likely to be interested in:

- Data management
- Record linkage
- Pattern elucidation/aggregate analysis

Each of these functions is a goal that can be achieved through shaping of the database in the design process, and each will require some elements of the database design to be conducted in specific ways, although they are by no means mutually exclusive. And this latter point is an important one, given that most historians will want to have access to the full range of functionality offered by the database, and will likely engage in research that will require all three of the listed types of activity. Or, to put it another way, many historians are unlikely to know precisely what it is they want to do with their database at the very beginning of the design process, which is when these decisions should be taken. This is why, as we shall see later in this section, many historians are inclined to design databases which maximise flexibility in what they can use them for later on in the project (a goal which will come at the price of design simplicity).

The data management aspect of the database is in many cases almost a by-product of how the database works, and yet it is also one of its most powerful and useful functions. Simply being able to hold vast quantities of information from different sources as data all in one place, in a form that makes it possible to find any given piece of information and see it in relation to other pieces of information, is a very important tool for the historian. Many historians use a database for bibliographical organisation, allowing them to connect notes from secondary reading to information taken from primary sources and being able to trace either back to its source. The simpler tools of database software can be used to find information quickly and easily, making the database a robust mechanism for holding information for retrieval.

Record-linkage is where the database, and particularly where the relational database (see Sections D and E), comes into its own. Connecting people, places, dates, events and themes across sources, periods and geographical or administrative boundaries is clearly an incredibly useful task to perform, and whilst the database can do this, the efficiency and accuracy of the linkages will be dictated by both the design of the database structure and the nature of the data model (see Section E).

Finally once the information from your sources has been converted into data, the database software can be employed to group information together. Once records can be aggregated, then it becomes possible to count them, meaning that statistical analyses can be performed and structural patterns can be identified within the information. Again, however, the efficiency and accuracy of this kind of function will depend on the design of the database and the manner in which the information has been converted. In particular, this kind of functionality will depend a great deal upon the latter, and if the historian aims to perform this kind of analysis extensively, then there will need to be a considerable effort put into applying a ‘standardisation layer’ to the data (see Section C4).
C3. Conceptual models of database design

Whilst it is true that every database ever built has been designed specifically for a particular conjunction of purpose and data, and is therefore to a greater or lesser extent distinctive, it is also true that there are two principal overarching approaches to designing databases. The two conceptual models are known as:

The Source-oriented approach (sometimes called the Object-oriented approach)

and

The Method-oriented approach (also known as the Model-oriented approach)

These two models should be viewed as polar opposites at the ends of a sliding scale, where the design of a database is based on an approach somewhere between the two extremes. Every database design will be something of a compromise, and no database will ever constitute the ‘perfect source-oriented database’, nor will there ever be the ‘perfect method-oriented database’.

![Diagram of the Source-oriented approach](image)

C3i – The two conceptual approaches to database design

The Source-oriented model of database design dictates that everything about the design of the historical database is geared towards recording every last piece of information from the sources, omitting nothing, and in effect becoming a digital surrogate for the original. The information contained within the sources, and the shape of that information, completely ordains how the database must be built.

The lifecycle of an ideal source-oriented database can be represented thus:

![Lifecycle of the Source-oriented database](image)

C3ii – Lifecycle of the Source-oriented database

This approach to database design is very attractive to the historian as it places the sources at the centre of the database project. Entering data into a database is a very time consuming activity, however, and this becomes much more so if you are taking pains to record all of the information that exists in your sources. Ultimately you will need to make choices about which information you will exclude from the database, contrary to the principles of the Source-oriented model, which will undermine the database’s role as a digital surrogate for your sources but which will at least allow you to perform your research within a reasonable period.
The Source-oriented approach, if rigidly applied, can lead to a design that quickly becomes unwieldy as you try to accommodate every last piece of information from your source, some of which may only occur once. But, it does allow for wider analytical approaches to be taken later, so that potential queries are not reliant on the initial research agenda, meaning that the database does not restrict the directions your research might take. It also allows you the reassurance of not having to anticipate all of your research questions in advance, which the Method-oriented model does. The Source-oriented model transfers the source (with all its peculiarities and irregularities) in a reasonably reliable way into the database with little loss of information – ‘everything’ is recorded (or at least what is excluded is done so by your conscious choice), and if later something becomes interesting, you will not have to go back to the source to enter information that you did not deem interesting enough to begin with. The Source-oriented model also enables you to record information from the source ‘as is’, and lets you take decisions about meaning later – so ‘merc.’ can be recorded as ‘merc.’, and not expanded to ‘merchant’ or ‘mercier’ at the point of entry into the database. [1]

At the other end of the scale, the lifecycle of the Method-oriented model database could be represented in a different way:

![C3iii – Lifecycle of the Method-oriented database](image)

This approach to database design is based on what the database is intended to do, rather than the nature of the information it is intended to contain. Consequently, if adopting this model for designing your database, it is absolutely vital that you know before you begin precisely what you will want to be able to do with the database – including what queries you will want to run. The level of precision needed here should not be underestimated either, given that the database requires a high degree of granularity to perform analysis – the database will not be able to ‘analyse the demographic characteristics of the population’, for example, whereas it will be able to ‘aggregate, count and link the variables of age, gender, marital status, occupation, taxation assessment, place of residence’ and so on. When designing any database it will be necessary to think at this latter level of detail, but if you are designing a Method-oriented database then it becomes much more important.

Method-oriented databases are quicker to design, build and enter data into, but it is very hard to deviate away from the designed function of the database, in order to (for example) pursue newly discovered lines of enquiry.

Ultimately, historians will need to steer a middle course between the two extreme models, perhaps with a slight tendency to lean towards the Source-oriented approach. When making decisions about what information you need from your sources to go into the database, it is important to take into account that your needs may change over the course of a project that might take a number of years. If you want to be able to maintain the maximum flexibility in your research agenda, then you will need to accommodate more information in the database design than if you are very clear on what it is you need to do (and what that is will never change). If you do not know whether your research needs will change, err on the side of accommodating more information – do not exclude information about servants unless you are absolutely sure that you will never want to treat ‘households with servants’ as a unit of analysis, because if you have not entered that information, then it will not be there to query later on.

However you should not dismiss the Method-oriented model out of hand when considering the approach to your database design. If you know your source(s) very well in advance, and you have definite predetermined research needs, and you know you will not be attempting to recover all the information from
the source, and you know in advance exactly how you will treat your data and what questions you will ask of it – if all this is true, you can use the Method-oriented approach. Alternatively, if you are creating a database which is not actually for historical research, but is designed to be a resource with pre-defined functionality and a limited set of tools that a user can use,[2] then a Method-oriented design is also appropriate.

[1] Leaving this kind of ‘normalisation’ until later in the project is beneficial as it allows you to make decisions about the meaning of data until you have the full body of data to act as context.


DHSI 2018 -- the rest of "Designing Databases For Historical Research" is optional reading.
C4. Database ‘layers’

Databases often involve several stages of work before they can be fully utilised for analysis. This is because well designed databases arrange data into several layers. The ‘Three Layer’ model of database design serves to illustrate how the organisation of different types of data within a database can dramatically improve the analytical potential of that database. The Standardisation Layer in particular is one that historians should invest time and effort into developing, and practical methods of doing this will be addressed in Section F.

C4i –The ‘Three Layer’ model of database design

The basic premise of the Three Layer model is to create different kinds of data. The first kind is data that is derived from the source and which is entered into the database in the form in which it appears originally; the second kind is data that is adapted to obey more closely the strictures imposed by the database to enable the retrieval, querying and analysis processes to be performed more efficiently and accurately. This second type of data is standardised in one manner or another, where for example variations of spelling of a particular entity are codified into a single form to make it easier to find them in the database.

Before we discuss the specific layers, it is important to make the point that you will need to keep the different layers separate: that is, you should always be able to tell whether a piece of data is from the source or whether it has been standardised in some way by you. In terms of database structure, every field will always belong to one layer only, although tables can contain fields that belong to more than one layer.

The Source layer

This layer comprises the tables and fields (see Section C5) that contain information taken from the source and the forms of the data which have been entered verbatim from the original. No adaptation of the original information has taken place in this layer. This allows us to retrieve information that shows what the source actually said.

The Standardisation layer

This layer comprises the tables and fields which contain data that you have adapted to make analysis easier, and will include data where spelling has been standardised, where abbreviations have been expanded, where a single dating system is used and so on (see Section F). There are two opportunities for
creating this layer, either at the database design stage, or after all the data entry has been completed. If the
former approach has been chosen, then during data entry you must be rigorously consistent in the way
that you enter your standardised forms (e.g. always spelling ‘John’ in the same way), and you must
document how you have standardised. If done after data entry as post-processing, you can create your
standardised values globally across the whole body of data, but this can be time consuming when dealing
with lots of information that needs to be standardised. If possible, the former approach is almost always
the better option to take.

The Interpretation (also known as Enrichment) layer

This layer is in many ways an optional one, whereas the other two are not. This layer comprises data and
materials which have been drawn into the database from elsewhere other than your sources, in order to
make the database and the data it contains a more useful and powerful resource. This can consist of
classification, interpolation, interpretation and derived variables. It can also include making links between
data: two pieces of information when linked make a third new piece of information. It can include
reshaping the data in the database to increase usability (for example drawing together dozens of records
about an individual in order to create a single encyclopaedic record which contains everything there is to
know about that person), and it can include incorporating contextual material drawn from entirely
separate (perhaps secondary) resources. Many databases used for personal research do not include an
Interpretation layer.

Describing these layers in this way might suggest that they each exist in entirely separate ‘areas’ of the
database as discrete collections of data. In most cases this is not true, however. In most cases data
belonging to each field will co-exist within tables, but within separate fields within the tables (see Section
C5): for example you might create two fields for ‘occupation’ in the same table that records information
about people, in one field (belonging to the Source layer) you can record how the occupation is presented
in the source, in the second field (belonging to the Standardisation layer) you can record a standardised
version of the occupation. The standardised version will be used for querying and analysis, because it will
be easier to find (by virtue of the fact it is always represented in the same way).

When moving between layers it is important you are always aware of what belongs to the Source layer
and what belongs to the other layers. This will usually be obvious (at least to you, if not to others), but
there should be explicit rules defined (and recorded) explaining the demarcation between the layers. In
other words the layers should be managed so that it is always possible to backtrack or undo a piece of
interpretation that appears in the database in the event of you changing your mind about something.
Similarly you should always be able to get from the result of a query which draws upon standardised data
back to the original source data. Much of this layer management will occur through the structure of the
database and will not require active intervention on your part, but it is worth remembering at all times that
any given piece of data belongs to only one or other of your layers.

These are the principles of database layers, but as with everything else about database design, there is a
degree of flexibility about how steadfastly you hold to them. For example, employing a source layer and a
standardisation layer can mean entering lots of data twice in two different forms – the original version of
an occupation, and the standardised version, for example – which will clearly slow down the data entry
process. There is a judgement to be made by every historian creating a database as to how far these
different layers should be used, and specifically, whether or not some pieces of information need both a
source layer and a standardised layer. If you are unlikely to use a piece of information for analysis, then
perhaps it does not need to have a standardised version; if you do not need to have a piece of information
in its original form, then perhaps its source layer version is not required. These decisions need to be made
carefully, as they impact directly upon how long your data entry will take and how easily you will be able
to perform your analysis.
C5. Database definitions – tables, fields, records, values, rules and datatypes

This section of the Handbook will introduce a few definitions and database terms which will act as a shorthand for various concepts and processes that will be discussed in other sections. Again, the issues discussed here will apply generally to all databases designed for historical research, as well as to most databases in general.

Harvey and Press provide a definition of a database:

“A database is a collection of inter-related data organised in a pre-determined manner according to a set of logical rules, and is structured to reflect the natural relationships of the data and the uses to which they will be put, rather than reflecting the demands of the hardware and software.”

For the full description see: Further Reading

which is a useful if perhaps formal way of describing both the content and environment of a database. Within the database itself, however, are a number of different ‘things’, called Objects,[1] which serve a variety of functions (these include tables where the actual data is stored, queries, user interfaces and so on). For the moment we will concentrate on only the first of these objects, the tables, and we will look at some of the terms connected with them.

There are four main elements to any table in a database, and each of these have (somewhat inevitably) a number of names:

§ Table (also known as Entities)
§ Field (also variously known as Column, Variable, Attribute)
§ Record (also known as Row)
§ Field name (also known as the Attribute name)

[C5i –The database table, showing fields and records]
In each database, data are stored in tables, and most databases will have more than one table. These tables will have relationships between them which connect the information they contain and will thus be ‘relational’ (see Section D). Tables are made up of fields (columns) and records (rows). Each field contains one type of information, and for each record the information in that field will be of the same kind.

Database Rules

Returning to the ‘logical rules’ referred to by Harvey and Press, these can be summarised as follows:

1. The ‘information rule’: all information in a relational database is represented in one way – as a value in a field in a table
2. Each table in a relational database will have a name which uniquely identifies it and which will distinguish it from the other tables; the table should contain information about only one entity.
3. Each field within a table is given a unique name within the table.
4. The values in each field must have the same logical properties, that is, must be of the same datatype: numerical or text (we will be looking at datatypes shortly)
5. Records will contain closely related information about a single instance of the entity (that is, the subject of the table – for example, the forename and surname of a single individual in a table about individuals)
6. The order of records in a table must not be significant
7. The order of fields in a table must not be significant
8. Each complete record must be unique
9. Each field should contain ‘atomic values’: this is, indivisible data (e.g. first and last names of an individual should always be held in separate fields)

In many cases the database software will prevent you from breaking these rules, but in some cases it is possible to circumvent them, whether intentionally or otherwise. With those rules that can be broken, it is almost always unwise to do so, as it will generally lead to confusion (for the database if not its user) when it comes to performing analysis.

Datatypes
We have seen that databases are made up of tables, and that tables follow certain rules. One of the rules to be considered is that each field within a database must be of a certain datatype. There are a number of different datatypes, and although the names by which these datatypes are known will vary slightly from database software to database software, what they do and how they operate will be essentially the same.

In essence what datatypes do is to control what type of data is entered into a field. Each field in each table is assigned a datatype, usually ‘text’ or ‘numeric’, and this in turn dictates what kind of data can be entered into that field. The purpose of datatypes is partly to act as a kind of quality control mechanism, to help prevent data being entered incorrectly; and partly they are used to help the database understand the meaning of what is typed in. For example, the piece of data ‘23/03/2011’ will not be recognised as a date by the database if it is entered into a field that has the datatype ‘text’, whereas it will understand the meaning of this value if it is entered in a field with the ‘date/time’ datatype.

A brief description of the principal datatypes follows.

The basic datatypes: text

- This is the default datatype for your fields which will be assigned by the database whenever you add a new field to a table
- This datatype will allow the entry of textual and numerical values, although it will treat that latter differently from numerical values entered into a ‘number’ datatype field
- Fields with this datatype will generally allow a maximum of 255 characters to be entered

The basic datatypes: memo

- Fields with this datatype are used for lengthy texts and combinations of text and numbers
- Up to 65,000 characters can be entered (the precise number may change depending on the database software being used)
- Data in these types of field cannot be sorted
- Data in these types of field are difficult, although not impossible, to query

The basic datatypes: number

- This datatype allows the entry of numerical data that can be used in arithmetical calculations
- There are a number of variations of this datatype, which control aspects of the numbers that can be entered, such as the size of the numbers, the number of decimal places and so on:
  - Byte: Stores numbers from 0 to 255 (no fractions)
  - Decimal: stores positive and negative numbers down to 28 decimal places
  - Integer: Stores numbers from –32,768 to 32,767 (no fractions)
  - Long integer: (default) stores numbers from –2,147,483,648 to 2,147,483,647 (no fractions)
  - Single: stores very large positive and negative numbers to 7 decimal places
  - Double: stores very large positive and negative numbers to 15 decimal places

The basic datatypes: date/time

- This datatype enables the entry of dates covering the period 100 through to 9999
- This datatype can be customised in order to control the format of dates that are entered
- Warning: in Microsoft Access, the Office autocorrect feature may well change some of your dates if it is active (e.g. “02/12/04” will autocorrect to “02/12/2004” unless you enter the year in full)
This datatype will allow the sorting of records chronologically, which the same values entered into a text datatype field would not (the latter would sort the records alphabetically – alphabetically “01/01/2010” would come before “31/12/1245”)

The basic datatypes: currency

- This datatype allows the entry of numerical values data used in mathematical calculations involving data with one to four decimal places, with the inclusion of currency symbols

The basic datatypes: autonumber

- This datatype automatically generates a unique sequential or random value whenever a new record is added to a table. Further information
- AutoNumber fields cannot be updated, that is, you cannot enter data into them manually

The basic datatypes: yes/no

- A field with this type of datatype will only contain one of two values (Yes/No, True/False, or On/Off)
- Quite often database software will represent this type of field as a checkbox in the table
- The underlying text value of a yes/no field will be -1 (for yes) or 0 (for no)

The basic datatypes: OLE

- A field with this datatype is one in which another file is embedded, as a Microsoft Excel spreadsheet, a Microsoft Word document, an image, a sound or video file, an html link, or indeed any other type of file

Every field in every table will need to have one of these datatypes assigned, and the decision as to which type is chosen should be factored into the database design process (see Section E). For most fields the datatype chosen will be either ‘text’ or ‘number’. Keep in mind how these two datatypes treat numerical data differently, particularly in terms of how they cause data to be sorted:

- 1,10,11,2,3,4,5,6,7,8,9 is how data will be sorted if the datatype is ‘text’ (that is, alphabetically)
- 1,2,3,4,5,6,7,8,9,10,11 is how data will be sorted if the datatype is ‘number’ (that is, numerically)

[1] For more on Objects please sign up to one of our face-to-face Database courses.

[2] Note in this table the Forename, Surname, Occupation, Residence and Comments fields contain values taken directly from the source, and are part of this database’s Source layer. The Stated_Sex field contains a standardised value (in fact a code), and is part of the Standardisation layer.
C6. Conclusion

All of the foregoing concepts and aspects of database organisation need to be taken into consideration when designing the database that will be used in your historical research, as each of them will either have an implication for how some of your historical information will need to be converted into suitably shaped data (see Section E), or else they will have an impact upon the types of linking, counting and analysing it will be possible to perform on your data. Designing the database will not only involve modelling the information from your sources, but it will also involve making decisions about the conceptual approach of your database, the functions you want it to perform, and how you want the tables and fields of the database to operate at the most minute of levels.
D. Relationships

D1. Introduction
D2. Functions of Relationships
D3. Types of Relationships
D4. Primary and Foreign Keys
D1. Introduction

As mentioned in Section C, most databases consist of data held in more than one table, and this is especially true for databases where the data is derived from historical sources. Relationships are created between the tables to connect the data in one to the data in the other: more precisely, relationships are used to connect specific records in one table to specific records in another. In many ways relationships, and the whole relational data model, comprise the most difficult aspect of designing a database, and not necessarily because they are difficult to actually create. What is difficult about relationships is why we need them – the reasons for using related data can seem obscure and unnecessary at the start of a database project, especially if you have limited experience of using databases. They are, however, extremely important. In essence what relationships allow us to do is two-fold: firstly they allow us to simplify very significantly the process of data entry (and incidentally at the same time enable us to protect the quality of the data we enter by limiting data entry errors); and secondly they serve to ensure that the results of our queries are accurate by making it clear precisely what it is that is being queried.
D2. Functions of relationships

These dual functions of relationships are best illustrated with an example.

Imagine a database which contained data about people and the hats that they owned, comprising personal information about name, gender, age and so on, as well as information about hat type and the materials used in their manufacture. There are two ways that this database could be designed:

- A single (flat file) table data model, where all of the information about people and hats was entered into the same table
- A relational data model where two tables are created, one to contain information about people, and one to contain information about hats

The two scenarios are both feasible and will allow you to conduct a detailed analysis of people and their hats, but each brings with them some very significant consequences if chosen.

Scenario A: all information in one table

Scenario B: two related tables
D3. Types of relationship

Once the need for related tables arises, it is important to understand that there are different kinds of relationship that can exist between two tables. These differences are not technical, in the sense of being dictated by or a product of the mechanisms of the database; they are instead a function of the logical, semantic connection between the information between the two tables.

There are three types of relationship that can exist between two tables in a database, not all of which are useful or desirable.

One-to-one relationships:
- This relationship exists where a record in Table A can only have one related record in Table B, and a record in Table B can only have a single matching record in Table A.
- For example, an MP can have only one constituency, and a constituency can have only one MP.

This type of relationship is unusual in a database, as in many cases where a one-to-one relationship exists, the information in the two tables could exist in a single table. This is not to say that if, as a part of your Entity Relationship Modelling process (see Section E), you discover a one-to-one relationship between two of your designed tables you should redesign the tables into a single one, only that you can if you want to. For example, if you wanted to, you could create a single table to enter information about Members of Parliament and about their constituency, as when it came to entering this data you would not encounter the duplication of information that was problematic in the example of people and hats in Section D2. On the other hand, tables are supposed to be discrete entities, and so logically speaking you might prefer to conceive of ‘MPs’ and ‘Constituencies’ as two different entities, and thus two different tables. The important thing to remember with one-to-one relationships is that the database software that you use to build your database will allow you to create this kind of relationship, and that it will not create any problems when it comes to running queries.
One-to-many relationships:
- This relationship exists where a record in Table A can have no, one or more matching record in Table B, but a record in Table B can only have one matching records in Table A
- For example, a mother can have more than one child, but a child can have only one biological mother

This is the most common type of relationship in found in databases, and is usually the type that you want to build into your designs. As illustrated in the people and hats scenario (Section D2) this type of relationship is used to overcome the kinds of problems that arise within the database when the information drawn from the sources would require the duplication of data if entered into a single table.

Many-to-many relationships:
- This relationship exists where a record in Table A can have no, one or many matching records in Table B, and a record in Table B can have no, one or more than one matching record in Table A
- For example, an author can write more than one book and a book can be written by more than one author

If you discover this kind of relationship operating within your database design at the end of the Entity Relationship Modelling process, then you have a problem which will need to be addressed before you can proceed to actually building the database. Many-to-many relationships will not work in databases, as they will essentially break any query you try to run on the tables related, throwing the query into a ‘loop’ which will generate gibberish as results.

Given how problematic this type of relationship is, it is somewhat disheartening to see how frequently they crop up when modelling historical information! The way of dealing with a many-to-many relationship requires something of a conceptual leap, as it requires the creation of a table, sometimes called a Junction Table, to sit between the two related tables. This Junction Table will act in an abstract fashion – the data it will contain will not be information as such, but they will serve to split the many-to-many relationship into two one-to-many relationships.

![D3ii - Many-to-many relationship between Author and Book tables](image)

Take the database which contains a table about Authors and a table about Books, which might be designed according to the Entity Relationship Diagram depicted in image D3ii (for Entity Relationship Diagrams see Section E2). The arrowheads indicate the ‘many’ side of a relationship, here indicating that
both tables are on the ‘many’ side, clearly highlighting a problem. To overcome the many-to-many relationship, we would insert a Junction Table to split the relationship into two one-to-many relationships, as indicated in image D3iii.

D3iii – Many-to-many relationship between Author and Book tables split with a Junction Table

Note that each record in the Junction Table contains three fields: a unique ID for each record (Junction ID), and then a field for each of the Author IDs and Book IDs. Each record therefore becomes a unique combination of Author and Book IDs, which indicates which books were written by which authors:

The Junction Table here is effectively circumventing the many-to-many relationship between books and authors, and each record it contains acts as a statement linking one or more author with one or more books. The first two records in the Junction Table, for example indicate that Author ID 1 was the writer of
Book IDs 1 and 2, whilst the last two records indicate that Book ID 9 was co-authored by Author IDs 2 and 5. The relationship between books and authors is managed by the Junction Table, whilst the details about books and authors are kept in their respective tables.

This arrangement, whilst somewhat convoluted, will enable the database to run queries that draw on information in both the Book and Author tables when it would otherwise not be able to due to the many-to-many relationship. It is therefore a very valuable technique to bear in mind when identifying relationships between tables as part of the database design process.
D4. Primary and Foreign Keys

Primary and Foreign Keys are absolutely crucial to the running of a relational database because they serve as the ‘anchors’ that keep relationships connected to the relevant tables. The keys are fields that serve a particular purpose within a table: they are not used to capture information drawn for the sources, but instead they are used to keep track of the information that is necessary for the database to know which records in one table are connected to records in the related table.

One of the ‘rules’ of database design (Section C5) states that ‘each complete record must be unique’, meaning that when taken as a whole, each record in a table must be unique. With historical sources, this can sometimes be problematic, as quite often the same information can crop up repeatedly over a period of time – such as, for example, when an individual is named as a witness to a number of wills across a decade. To make sure that the records of a table obey this particular rule of databases, it is necessary to guarantee that each record will be unique, and if the nature of our historical information prevents us from being able to guarantee this, we are forced to cheat. In doing so, we actually achieve a number of useful effects in the design of our tables.

The way that we guarantee that each record in a table will be unique is to add a field into which we can enter (or have the database enter automatically) a unique identifier, a value which will be different for every record that is added to the table. Normally this is a sequential number, such as the values in the various ID fields in the People and Hats database (Section D2). A sequential number value applied to each record that will be different for each record will guarantee by itself that every record as a whole will be unique – because the ID value will never be duplicated from one record to the next.

However this field which provides us with a unique value for every record also serves to act as the Primary Key field for a table, this being the field that acts as the anchor for the ‘one’ side of a one-to-many relationship. The field that acts as an anchor on the other side of the relationship that exists within the table that is on the ‘many’ side of the relationship is known as the Foreign Key. The Foreign Key field will not contain a unique value for every record: because it is on the many side of the relationships, the same ID value is likely to occur in more than one record. Both the Primary and Foreign Key fields contain the same information.

Consider the People and Hats database again:

Person table:

<table>
<thead>
<tr>
<th>PersonID</th>
<th>Forename</th>
<th>Surname</th>
<th>Occupation</th>
<th>Residence</th>
<th>Age</th>
<th>Gender</th>
<th>HairColour</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Samuel</td>
<td>Spade</td>
<td>Builder</td>
<td>Peckham</td>
<td>23</td>
<td>M</td>
<td>Blonde</td>
<td>£25,000</td>
</tr>
<tr>
<td>2</td>
<td>Philip</td>
<td>Marlowe</td>
<td>Accountant</td>
<td>Dulwich</td>
<td>35</td>
<td>M</td>
<td>Dappled</td>
<td>£34,533</td>
</tr>
<tr>
<td>3</td>
<td>Hercule</td>
<td>Poirot</td>
<td>Plumber</td>
<td>Chelsea</td>
<td>26</td>
<td>M</td>
<td>Bald</td>
<td>£250,000</td>
</tr>
<tr>
<td>4</td>
<td>Miss</td>
<td>Marple</td>
<td>Doctor</td>
<td>Surbiton</td>
<td>58</td>
<td>F</td>
<td>Grey</td>
<td>£1.50</td>
</tr>
</tbody>
</table>

Hat table:

<table>
<thead>
<tr>
<th>PersonID</th>
<th>HatID</th>
<th>HatType</th>
<th>HatColour</th>
<th>HatMaterial</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Baseball cap</td>
<td>Black/red</td>
<td>Polyester</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Bowler</td>
<td>Pink</td>
<td>Satin/fur</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Crash helmet</td>
<td>Green</td>
<td>Kevlar/fur</td>
</tr>
</tbody>
</table>
The relationship is one-to-many, a person can have many hats, and the relationship is anchored on the PersonID field which is present in both tables. The PersonID field contains the unique ID number that is assigned to every person who is entered in the Person table – because every person is only entered once in the Person table, the PersonID field in the Person table is unique, and it is the Primary Key. In the Hat table, because a person can own more than one hat, the Person ID value will not be unique – the same ID number will crop up every time a record contains a hat owned by that person (as is the case for persons 2 and 3). The PersonID field in the Hat table is therefore the Foreign Key.

When designing the database, there is a stage in the process where you will need to think about what fields will appear in your tables (see Section E2), and it is very important when doing so that you remember to identify the Primary and Foreign Keys for your tables. Every table you design should have a Primary Key field – a field with the datatype ‘autonumber’ which will generate a unique value for every new record you add. Not every table will have a Foreign Key field, only those that are on the ‘many’ side of a one-to-many relationship. When you do add a Foreign Key field to a table, remember that it will contain the same information (that is, the ID numbers) drawn from the field that is the Primary Key for that relationship. Without the Key fields, the database software will not be able to correctly manage the relationship, with the result that it will not be able to identify which records from the one table are connected to the which from the other, a situation that will make performing analysis or even simple retrieval of information all but impossible.
E. Entity relationship modelling

E1. Introduction
E2. Entry Relationship Modelling Step-by-Step
E3. Conclusion
E1. Introduction

Throughout this Handbook so far reference has been made to the translation and conversion processes involved in taking information from sources and turning them into data within the database. This section describes precisely the tasks involved in performing these processes, which are collectively known as Entity Relationship Modelling (ERM). The mechanics of ERM are in fact a lot less intimidating than the name implies, but it is nevertheless a complex activity, and one that is likely to prove challenging at the first few attempts. Luckily, however, the various stages of ERM draw very heavily upon the skills and experience that the historian utilises as a matter of course during their research anyway, which, unlike most aspects of database use, places the historical researcher at something of an advantage. The difficulty of the ERM process is directly proportional to the complexity of the source(s) being used in the research, with some types of sources being (relatively) simpler to model than others. Highly structured sources like census returns, lists of inhabitants, poll books and so on will be easier to model than ‘semi-structured’ sources such as probate inventories, which in turn will present fewer problems than completely unstructured material such as narrative texts and interviews, and so on. However all will have their own particular features and problems to complicate the modelling.

The process of ERM serves a number of purposes. Firstly, it makes the historian decide upon what it is the database is to achieve in terms of its functions. Secondly, it identifies the types of information that can be obtained from the sources, and in conjunction with the database’s chosen aims, aids the historian in deciding upon which information from the sources should be entered into the database, and which can be excluded. Thirdly, ERM makes the historian think in detail about the components of the database, its tables, fields, relationships, datatypes, and so on, decisions on all of which are crucial to a successful database design. Finally, it encourages the consideration of the layers of the database, what information needs to be entered into both the Source layer and the Standardisation layer, what can be entered only into the latter, and how extensive the latter needs to be. Once these tasks have been conducted, the historian is left with a very precise idea of what the database will look like, and, on a more practical note, will be left with the design of their database on paper (an Entity Relationship Diagram [ERD]).
E2. Entity Relationship Modelling step-by-step

Entity Relationship Modelling (ERM) is where you model the information from your sources to fit into a structure that will form the basis of your database design. The product of ERM is an Entity Relational Diagram (ERD), which will depict three components: entities (tables), attributes (fields) and the relationships between them.

Stage 1: Determine the purpose of the database

This stage is always the starting point of the ERM process, and is especially important if you are adopting the Method-oriented approach to your design, but it is a vital step in designing any database. No database exists without a purpose: with a Method-oriented database it is crucial that you know exactly what you want to do with your data before you design the database. At this stage you will need to make your decisions about what information you want to keep and what you want to discard, and you will need to be prepared to abide by the consequences of this decision throughout the lifecycle of your database project. This point deserves to be made emphatically: although it is always possible to retrofit the design of your database to include information that you had initially decided to discard, it is not always a trivial matter to do so, particularly if you have to enter another phase of data entry to collect the new information.

Stage 2: List entities

Once you know what you want your database to do, you need to divide your anticipated information into discrete subjects: each subject, or entity, will evolve into a separate table.[1] We split the entities into separate tables for the purposes of efficiency, for avoiding ambiguity, and because this allows us the maximum flexibility for querying once the data are in.

This stage of the ERM process sounds deceptively simple, but is in fact probably the most difficult step of the whole process, and it is certainly worth reading up on the subject (see the Further Reading section).

For example, consider a research project that was investigating political elections in eighteenth-century Bristol, and the sources consisted of a collection of poll books which recorded the votes cast by the local electorates in the communities where elections took place (image E2i):
E2i – An example page of an 18th century poll book[2]

With sources such as these we might pursue a research question which was something like: ‘Analyse the geographic and economic determinants of voting in early 18th century Bristol’. With a question like this we would be interested in geography, economic status and voting patterns in relation to a group of...
individuals. In terms of entities, we might conclude that there is only one: people, actually, more precisely, we would be considering voters, which would lead us to the position of deciding that we would need a table into which we would enter our information about voters.

Alternatively if we were using probate materials for our research, and wanted to create a database into which we could enter information extracted from wills, we would need to consider the entities from this source. We might conceive of our entities – our discrete subjects – as breaking down into ‘document’, with each will being treated as a different document; ‘person’, containing information about different types of people and their roles – testator, recipient, executor etc.; ‘bequest’ with a range of types of associated information; and ‘object’, being the object of a bequest. If our research was interested in the material culture of a period or place, this latter entity would be particularly important, whereas if the project was concerned only with networks of people and social interrelations, the ‘object’ entity might not be necessary.

In both these examples the possibility of these entities being identified has been stressed, because the definition of entities is as much a product of the purpose of the database as it is of the information contained in the sources. As mentioned at the beginning of this Handbook, no two databases will be designed in the same way, as no two databases will be built for the same purposes – different designers may well identify different entities based upon their unique appreciation of their research and their sources. And this is why to a large extent this step of the ERM process is the most difficult!

However it is perhaps worth considering the inclusion of three commonly chosen entities in the design of your database:

- People - with a related entity of ‘role’ (being the reason why they are present in the source)
- Document – where archival and bibliographical material can be entered (and thus enabling the tracking of every piece of data in the database to its source)
- Event – a slightly more abstract entity, one which describes an instance of whatever it is your source records (a trial, a taxation assessment, an election etc.) and where information about dates can be recorded.

Exercise

Assume you have a research project that is examining the demographic profile of families in a particular parish, and your principal source will be ecclesiastical registers of baptisms, marriages and burials. List the entities from this source that would need to be included in your database design.[3]

Stage 3: Identify the relationships between the entities

Bearing in mind the nature of relationships between different elements of information (see Section D), this step of the ERM process requires you to identify which of your entities are related, and what type of relationship exists between them. This is an exercise in abstract logic, and will take considerable practice: in addition, quite often, this stage will require revisiting stage 2 and redefining the entities you originally chose.

If we return to the database of wills mentioned in stage 2 with the entities ‘document’, ‘person’, ‘bequest’ and ‘object’, we would need to unpick the nature of the relationships between these entities. We might logically decide that the relationships would look something like this (the arrowheads depict the ‘many’ side of a one-to-many relationship (see Section D):
A single document (will) can contain information about more than one person, and also about more than one bequest, whilst a bequest can include information about more than one object, so all of these relationships are one-to-many.

Exercise

Identify the relationships between the entities chosen for the parish register database in Stage 2.

Stage 4: Investigate and rectify problems

This stage is fairly self-explanatory. It is possible to spot problems with the incipient design even at this relatively early point in the ERM process, and if they exist it is better to do so here than after investing work in the later stages.

Look out in particular for:

- Relationships which do not appear to be one-to-many: remember you cannot have entities related by a many-to-many relationship, and whilst you can have them related through a one-to-one relationship, it may be worth rethinking the two entities involved (see Section D)
- Redundant relationships: if entities can be linked in more than one way, you should work out which link should be kept and which should be discarded – if Table A is related to Table B, and Table B is related to Table C, then Tables A and C are already by definition related, and do not need a ‘direct’ relationship to exist between the two

Stage 5: List all attributes associated with each entity, and identify keys

This stage involves listing the precise attributes of each entity that has been identified in the previous stages of the ERM process, by deciding on the fields that should occur in each table. Each field is to contain one piece of information about the subject of the entity, and that one piece will be a single aspect of information that can be known about the entity (see Section C5). Once you have listed the attributes for each entity, you must then identify which fields will act as the primary and foreign keys in each table, remembering that relationships between tables do not exist between tables, but between a specific field in one table and a specific field in the related table (see Section D).
You should always as a matter of course add a generic ‘Notes’ field into each table, with the memo datatype, as it is likely to prove invaluable at moments of crisis when entering data (see Section F)!

Stage 6: Construct the Entity Relationship Diagram (ERD)

Once you have completed stage 5, you are in a position to create the ERD for your database design, which will resemble the heavily simplified example above (image E2iii).

Exercise

Construct an ERD for the parish register database, ensuring that you have identified your entities and their attributes, the relationships between the entities and the fields which act as keys.

[1] In fact they may evolve into separate groups of related tables, but for the moment this does not need to be considered.


[3] If you are not familiar with parish registers as sources, it is possible to find examples of transcripts (of varying quality) online. One such set of transcripts, for the county of Hampshire, can be found at http://www.knightroots.co.uk/parishes.htm (accessed 23/03/2011).

[4] Terms in black are entity (table names), those in red are primary keys, those in green are foreign keys and those in grey are the remaining attributes/fields, some of which pertain to the database’s Source layer, and some to its Standardisation layer. It is very important to note that this diagram is entirely illustrative, and is not intended to be prescriptive or definitive in any way!
E3. Conclusion

The process of Entity Relationship Modelling (ERM) is difficult, and rapidly becomes more difficult if you are blessed with a number of different kinds of sources, each of which contains rich information about a variety of subjects. If you are using multiple sources, it is a good idea to avoid creating entities that are source specific: for example, if you are using census returns and taxation lists, both of which contain information about people, do not create two tables for people (one containing the information from one source, the other from the second). Stick to the abstract logic of the information – what is important to your research is people, so accommodate all of the information about people in the same place. Not only does this make sense from the point of view of logic, but it will also make it much easier to find data about specific individuals later on (either manually or via queries): looking for a person is easier to do if everyone is located in one table rather than several.[1]

No Entity Relational Diagram (ERD) will ever be perfect, as with so much else involved in database design it will be a matter of compromise. The success of an ERD is something that can only be determined in one way – by the database performing the tasks it was intended and designed to do, and this is something that will not become evident until after you have begun entering data and using the database for analysis. This is why the creation of the ERD is (or should be) swiftly followed by a period of intense testing of the database ‘in action’, in order to quickly identify where the design is impeding the database’s purpose (see Section G).

[1] Ultimately of course this is a matter of personal judgement: you may decide that your entity is not ‘people’, but is in fact two separate entities comprising ‘census return’ and ‘tax payer’, in which case you would be able to argue for two separate tables. You would still face the problem of having to look for individuals in more than one table, however, should the need ever arise.
F. Problems facing the historian

F1. Introduction

F2. Problematic Information

F3. Standardisation, classification, and coding
F1. Introduction

This section of the Handbook is in some ways the most crucial, as it addresses a range of issues that affect the historian in particular during the course of designing their database. Most of these issues arise through the variability and ambiguity that inevitably accompanies the information present in historical sources, and which is not something which non-historical database creators encounter: usually these latter will know precisely what kind of information they will have, and can therefore comfortably predict how the database will need to be structured to make best use of it. Historians will not be able to predict the nature of their information so confidently, even if they know their sources intimately, as there will almost always be instances where the sources confound them – be it through extra information cropping up in the material which appears to be beyond the scope of the source, or through missing or illegible text, or through marginalia, deletions and so on.

This variability or ambiguity in the type and shape of information within the sources, (whilst one of the more exciting aspects of historical research), is not the only occasion where the materials and practices of the historian create difficulties when a database is brought into the equation. Chronology, topography, geography, orthography, and a range of other historical contexts introduce an element of ‘fuzziness’ into the data, and fuzziness is anathema to the strictures of the relational database model. The irregularity of historical information must be managed through the design of the database in such a way as to maintain a balance between keeping the detail and richness of the source to the extent required by the project, whilst at the same time applying enough standardising and reshaping of the information to allow the database tools to operate with maximum efficiency and flexibility.

At the root of most of these ‘interfacing’ problems is the fact that databases do not ascribe meaning to any piece of information you enter into it. The database does not recognise words or numbers or dates: when you enter data into a table, all the database understands is that a field contains a value which consists of a collection of characters organised in a particular order (something that is known as a ‘string’). When two strings are identical, such as might occur with the values appearing in the Surname field of a ‘Person’ table across two records, then the database understands that these two pieces of data are the same and have the same meaning. If two strings are not identical, then the database does not recognise them as meaning the same – even if to the historian it is clear that ‘Jno’, ‘John’ and ‘Johan~’ are all ways of representing the same piece of information.

For example (image F1i), when running a query in a database containing the names of members of a London Livery Company from the sixteenth to twentieth centuries, one would be able to search for everyone named John Smith:[1]
F1i – Query results of a search for individuals called John Smith (using standardised form of name)

As we see, there are 275 individuals called John Smith in the database. We only know this, however, because we chose to run the query using criteria in Sname field – the fields which contain the surname of individuals – and the StdFname field – the field which contains the standardised version of the forename. If we ran the same query using the criteria “John” and “Smith” using the non-standardised version of the forename (in the field Fname), we would retrieve a smaller number of records:
F1ii – Query results of a search for individuals called John Smith (using non-standardised form of name)

This time only 169 records are retrieved. The difference between the two sets of results arises from the variant spelling of the forename ‘John’ (which can be seen in the first image). When searched for ‘John’ in the standardised field, everyone called John had the same value; when we searched in the non-standardised field where the original spelling of the name had been entered, the record for anyone whose name was not spelt exactly as “JOHN” was missed by the query. Whilst the historian knows that all the variant spellings in effect mean the same thing, the database cannot recognise this so long as the strings are not identical. Thus, in order for our queries and analyses to be performed accurately by the database, we have to make sure that data which means the same thing can be identified by the database as meaning the same thing.[2]

[1] For detailed guidance on how to run queries in databases, please sign up to our face to face Database courses.

[2] The problem of non-identical strings not being recognised as meaning the same thing acting as an impediment to the running of queries can be mitigated to some extent through the use of various querying tools, but even then the problem can still lead to inaccurate query results.
F2. Problematic information

There are certain categories of historical information which are habitually problematic, and unfortunately these tend to be those subjects that often constitute analytical units, namely geography, chronology and orthography.

Geographical information

The problem with geographical information as it occurs in historical sources is that the boundaries of administrative units overlap and change over time, so that the same physical location can belong to different counties/parishes/wards/precincts and so on depending upon the date of the source being consulted. Obviously this means that if your sources cover a long period of time, you will need to be aware of what implications of any boundary changes in that period may have for your data. This is especially true if you are recording data in a hierarchical fashion: for example if you have a field in a table for ‘Parish’, and another for ‘County’, and every record will be given a value in each field. If the parish of St Harry Potter is situated in the county of Hogwartshire at the beginning of the 17th century, then records connected with this parish would have these two values entered into the respective fields in the table. If, however, administrative changes in the 18th century alter the county boundaries so that St Harry Potter suddenly belongs to the county of Elsewhereshire, then the records will have the values of St Harry Potter in the parish field, and Elsewhereshire in the county field. Whilst this is accurate, it suddenly causes a problem for the database, in that you will have a number of records with the same string in the ‘Parish’ field – and so will be recognised by the database as meaning exactly the same thing - but which historically speaking have different meanings at different points in time.

In this instance there are two ways of dealing with this issue. Firstly, you simply stay aware of the problem, and when running queries on parishes you take the ‘County’ field into account as well as the ‘Parish’ field. This will enable you to specify which version of the parish of St Harry Potter you are analysing. Secondly, you could modify the Parish value to specify which version it is, so instead of entering St Harry Potter, you could enter St Harry Potter: Hogwartshire or St Harry Potter: Elsewhereshire into the Parish field. This would simplify the complication of running queries in this situation, but it would technically break the database rule about ‘atomic values’ (see Section C5, Rule no.9).

This particular problem is even more significant when it is not just the geographical boundaries that change, but when the actual entities themselves change. For example, 17th century London had over 100 parishes in the early part of the century, many of them absolutely tiny in terms of area and population. After the Great Fire, the opportunity was taken to rationalise the parishes, with the result that many were merged or united, often with the newly created entity retaining the name of one of the pre-Fire parishes, whilst each parish still maintained its own existence for some administrative purposes (eg. St Martin Ironmonger Lane and St Olave Jewry). Here the problem is not one of changing hierarchy (which parish belongs to which county), but one of meaning (what area/population is the source referring to at this date when referring to ‘St Martin Ironmonger’?). Various approaches to solving this are used, including that for the preceding example, but what is most important is to be clear in the data at all times precisely what is meant by the geographical terms you enter into the database.

Chronological/dating information

All of the possible problems created by shifting geographical terminology apply to the identification of dates in historical data. This is clearly a more serious issue the further back in history your sources were generated, when calendars and dating systems were more varied and plentiful, and record-keepers had
more of a choice in what dating system they could choose. The important thing to remember here, as with geography (and indeed everything else entered into the database), is that the database does not recognise meaning. The database will have no concept of when the ‘Friday after the Feast of the Decollation of St John the Baptist in the thirty-first year of Henry III’ was,[1] which means that this date, as a value, cannot be treated chronologically by the database (that is, sorted or queried by date). Regnal years, mayoral years, feast days, the days of fairs and markets etc. when used to date information in the sources will need to be converted into a value that uses an actual modern date format. Alongside this there is of course the issue of the shift from Julian to Gregorian calendars, so that if your data spans 1752 you will need to convert years into one of the Old or New Style systems.[2]

Do not forget the datatype of the field into which dating information will be entered (see Section C5), bearing in mind that ‘Text’ datatype fields will sort dates alphabetically whereas ‘Date/Time’ datatype fields will sort them chronologically.

Orthography/variant forms

This is the really big area in which historical sources provide information that is problematic for the database: how do you deal with information that appears with many different spellings or in entirely different forms when in reality it means the same thing (or at least you wish to treat it as the same thing)? How will you deal with contractions and abbreviations, particularly when they are not consistent in the source? How will you accommodate information that is incomplete, or is difficult to read or understand where you are uncertain about its meaning? All of these issues are practically certain to crop up at some point in the data entry process, and all of them will need to be addressed to some extent to prevent problems and inaccuracies arising during the analysis of your data (for the impact that these issues have upon querying, for example, join one of our face to face Database courses.


[2] Note that this does not necessarily literally mean ‘convert’: it would be entirely reasonable if your research required it to have two fields to enter date information, one that contained the date verbatim from the source, and the second into which the modern rendering could be entered. Querying and sorting could then take place using the latter field.
F3. Standardisation, classification and coding

The principal way forward for accommodating data containing these kinds of problems is to apply (often quite liberally) a standardisation layer into the design of the database (see Section C4) through the use of standardisation, classification and coding. These three activities are a step removed from simply gathering and entering information derived from the sources: this is where we add (or possibly change) information in order to make retrieving information and performing analysis easier. We use these techniques to overcome the problem of information that means the same thing appearing differently in the database, which prevents the database from connecting like with like (the fundamental pre-requisite for analysing data). For historians this is a more important step than for other kinds of database users, because the variety of forms and ambiguity of meaning of our sources does not sit well with the exactitude required by the database (as with the example of trying to find all of our records about John Smith, Section F1), so that more of a standardisation layer needs to be implemented.

Standardisation, classification and coding are three distinct techniques which overlap, and most databases will use a combination of the three when adding a standardisation layer into the design:

Standardisation

This is the process of deciding upon one way of representing a piece of information that appears in the source in a number of different ways (e.g. one way of spelling place/personal names; one way of recording dates and so on) and then entering that standardised version into the table. Consider using standardisation when dealing with values that appear slightly different, but mean the same thing - ‘Ag Lab’ and ‘Agricultural Labour’ as values would be treated very differently by the database, so if you wanted them to be considered as the same thing, you would signal this to the database by giving each record with a variant of this occupation the same standardised value.

Classification

This is the process of grouping together information (‘strings’) according to some theoretical, empirical or entirely arbitrary scheme, often using a hierarchical system in order to improve analytical potential. Classification is about allocating groups, and then placing your data in those groups. These groups can be hierarchical, and the hierarchy will let you perform your analysis at a variety of levels. Classification is less about capturing the information in your sources and is much more about serving your research needs.

When using a classification system it is very important to remember two things: firstly, since it is an arbitrary component of your database’s Standardisation layer designed to improve your research analysis, the system does actually need to meet your has to be able to meet the requirements you have for it. Secondly, therefore, the system needs to have been devised before data entry begins, it needs to intellectually convincing (at least as far as your historical methodologies are concerned) and it needs to be applied within your data consistently.

It is also worth being aware of how other historians have classified their information. There have been many classification systems created by the good and the great of the historical profession,[1] many of which have been used subsequently by others for two reasons: they allow comparability between the findings of different projects; and because they allow historians to turn different sources into continuous series of information. That is, two projects investigating the same thing at different periods may have to rely on different sources: by classifying their (probably slightly different information) into similar classification systems, a case can be made (convincingly or otherwise) that the research is comparable. This is not to say that you should necessarily try to adopt an existing scheme rather than develop one that
suits your research better, but it is worth keeping in mind if you are interested in comparing your analysis with that of another historian. In addition, given that classification systems in practice really only entail adding an extra field in a table into which the classified value is added, there is nothing stopping you (other than perhaps time) from employing more than one classification system for the same information in the database.

A detailed example of a classification system can be seen in an ongoing project which is investigating the material aspects of early modern households, and which uses a database to record minutely detailed information about material objects. One of the many ways it treats the information about objects is to classify objects by type, in order to be able to compare like objects despite the often substantial differences in the ways they are referred to in the sources. This works by adding a field in the table where item type data is recorded into which an ItemClass code value can be added:

F3i – Data about material objects that have been classified and coded

The ItemClass field here is populated with codes, and these codes record precisely what type of item the record is about (you can see what the source calls the item in the ItemDescr field).[2] The fact that the code is a numeric value, and the fact that the same numeric code is applied to the same type of object regardless of how it is described in the source, means that the ItemClass field acts as a standardised value.

Additionally, however, the ItemClass field enables the use of a hierarchical classification system (to examine a partial sample of the classification system, download the Microsoft Excel file Material Object Type Classification.xls ). The hierarchy operates by describing objects at three increasingly detailed levels:

§ Code I: the broadest level (for example, Linen (household); Storage; Tools; Clothing – Outer; Lighting etc.)

§ Code II: the middle level, offering sub-groups of Code I (for example Tools > Domestic implements; Clothing – Outer > Footwear)

§ Code III: the most detailed level of description (for example Clothing – Outer > Footwear > Boots)

To illustrate this we can take the example of how the database classifies objects that are used for seating:

F3ii–Classification system for objects in the category of ‘Seating’

You will notice from the Microsoft Excel spreadsheet that each code level has a two or three digit numeric code, so Code I: Seating has the numeric code 05, that for Code II: Chair is 02, and that for Code III: Wicker Chair is 006. These individual codes become elided into a single numeric code (in the
case of the wicker chair – 0502006) which is the value that gets entered into the relevant single field (ItemClass) in the record for the wicker chair in the database.

This may sound complicated and slow to implement, but the benefit of doing so is considerable. Firstly, the database can be created so that the codes can be automatically selected rather than memorised by the database creator, so that they do not have to stop to remember or look up what code needs to be entered for any given object. Secondly, and here is the principal reason for employing a hierarchical system, once the data have been coded, they can be analysed at three different semantic levels. The historian could, if they wished, analyse all instances of wicker chairs in the database by running queries on all records which had the ItemClass value “0502006”. Alternatively, if they were interested in analysing the properties of all the chairs in the database, they could do so by running queries on all records with an ItemClass value that begins “0502***”. Lastly, if the point of the research was to look at all objects used for seating, a query could be designed to retrieve all records with an ItemClass value that began “05*****”. This is an incredibly powerful analytical tool, and one that would be impossible to achieve without the use of a hierarchical classification system: to run a query to find all objects used for seating without a classification system would require looking for each qualifying object that the historian can anticipate or remember, by name and taking into account the variant spellings that might apply.[3]

Hierarchical classification systems are very flexible things as well. They can include as many levels as you require to analyse your data, and they do not need to employ numeric codes when simple standardised text would be easier to implement.[4]

Coding

Coding is the process of substituting (not necessarily literally) one value for another, for the purpose of recording a complex and variable piece of information through a short and consistent value. Coding is often closely associated with classification, and in addition to saving time in data entry (it is much quicker to type a short code word than it is to type five or six words) codes additionally act as standardisation (that is, the same form [code] is entered for the same information no matter how the latter appears in the source).

These techniques are implemented to make the data more readily useable by the database: the codes, classifications and standardised forms which are used are simple and often easier to incorporate in to a query design than the complicated and incomplete original text strings that appear in the source; but more importantly, they are consistent, making them much easier to find. However there are a number of things to bear in mind when using them, the most important of which is there are two ways of applying these techniques:

- By replacing original values in the table with standardised/coded/classified forms
- By adding standardised/coded/classified forms into the table alongside the original values

Both of these approaches present a trade-off between maintaining the integrity of the source and improving the efficiency of the potential analysis, in much the same way as the choices offered as part of the design process when selecting the Source- or Method-oriented approach to the database (see Section C3). The first approach to standardising, to replace the original version of source information in any chosen field(s) with standardised forms of data, enables the speeding up of data entry at the expense of losing what the source says. It also serves as a type of quality control, as entering standardised data (especially if controlled with a ‘look-up list’) is less prone to data entry errors than the original forms that appear in the source.
The second approach, to enter standardised values in addition to the original forms, allows for the best of both worlds: you achieve the accuracy and efficiency benefits of standardisation without losing the information as it is presented in the source. Of course, this happens at the cost of extra data inputting time, as you enter material twice.

When considering both approaches, bear in mind that you will only need to standardise some of the fields in your tables, not every field in every table. The candidates for standardising, classifying and coding are those fields that are likely to be heavily used in your record-linkage or querying, where being able to identify like with like values is important. Creators of databases built around the Source-oriented principle should exercise particular caution when employing these techniques.


[2] Note in passing that many of the other fields in this example contain codes as well – this table contributes substantially to the database’s Standardisation layer.

[3] It would, for example, need to look for all stools, buffet stools, wicker chairs, forms, settles, benches etc., leading to extremely complicated queries with possibly more criteria that the database can handle. For criteria in queries please sign up to one of our face-to-face Database courses.

[4] Indeed numeric codes are somewhat old fashioned in modern database usage, although they are no less efficient for being outmoded.
G. Conclusion

If you apply the principles and techniques discussed in the design of your database, you may well find that you spend a considerable amount of time in the process. Unfortunately there is no getting around this: designing databases simply is a time consuming business, especially if you have adopted a Source-oriented approach and you are working with a range of different, rich and complex sources. However, the time you spend working on the design will be more than repaid when it comes to the data entry and data analysis stages of the database project – and this cannot be overstated. Historical sources will give rise to all manner of complications and problems, intellectual and in terms of the mechanics of databases, and the more you can anticipate these and accommodate them in the design of the database, the more efficient and less frustrating the subsequent use of the database will be.

Before you begin the process of designing your database, and producing your Entity Relationship Diagram, it is worth spending a little time seeing how other historians have designed their databases (see the resources listed in the Further Reading section). You should also read through the other HRH Handbooks on Databases by Mark Merry as these describe in detail the processes of building databases and performing analysis respectively; and it completely necessary to see what is required in order for these processes to work smoothly, so that the design can facilitate take these requirements into account from the very beginning.

Finally, it is worth reiterating that designing databases is difficult, and there is no substitute for practice. No database is ever perfect, and the only indicator of quality, or success, when it comes to database design is whether or not it serves the various functions that you intended. If you can manage the information from your sources in the way that you need, and if you can perform the analysis that you require, and if you can be as flexible as you need in both of these areas, then your design is successful. But you do not have to wait until the latter stages of your database use to find out how successful you have been in the design – you can and should test the design of the database very early on. After producing your Entity Relationship Diagram, build a structural prototype of your database (that is, with only the tables and relationships, without worrying too much about the other tools that go into creating the database application) and spend a week entering data. If you are using multiple sources, enter material from each of the sources. As soon as you start entering data you will very quickly begin to see where any deficiencies in the design might be – look out for:

- Information that you would like to analyse which appears repeatedly, but you have nowhere specific to put it (i.e. for which you will need to add new fields)
- If you find yourself repeating information from record to record, you will need to think about re-ordering your relationships to prevent this (see Section D)
- Watch out for your datatypes, and change them where they are unhelpful
- Look for data that could be standardised or classified
- Look out for information that you had not anticipated when designing the database

It is likely that you will find examples of all of these in a very short space of time. Once you have spent some time entering data, design and run some queries to test whether or not the research questions you know you will want answers to can actually be answered by the current design. Running queries is the ultimate test of whether the database design works or not, and it is likely that you will find yourself rearranging fields in the light of what you learn. The queries will also highlight (often starkly) how much standardising of information you will need to engage in.

Once you have finished this testing, and moved on to design and rebuild ‘Version 2’ of the database, you will be well on the way to creating one of the most powerful research tools available to the historian. It will be a struggle to begin with, but it will be worth it in the end!
Sometimes I start class discussions by comparing image quilts of Google searches for “digital” (left) and “humanities” (right).

Today I’d like to talk about the ways in which humanists think about data, and how that’s distinct from the ways in which scientists and social scientists think about it.

Even though I think our issues can be pretty different, I want to make the case that there are some very promising ways in which libraries could make meaningful interventions in the humanities research lifecycle, both for what we might call traditional humanists and for digital humanists.

So I’ll start with what “traditional” humanists might need help with and then move on to the needs of what we call “digital humanists” (although I think in practice the distinction is a bit blurred).

I just want to say at the outset that there are people who specialize in humanities data curation, and I am not one of those people. A number of talented people, including Trevor Muñoz at the University of Maryland and Katie Rawson at the University of Pennsylvania, have started to take a very programmatic look at the data-curation needs of digital humanists. And I encourage you to check out their important work. But you don’t have Trevor or Katie; you have me! So what I can do is share my own perspective and experience on what it means to work with data as a humanist, and where libraries can help.
I’ll start with an anecdote, and I think that anyone who consults on digital humanities projects will be familiar with this scenario. Humanities scholars will sometimes describe elaborate visualizations to me, involving charts and graphs and change over time. “Great,” I respond. “Let’s see your data.” “Data?” they say. “Oh, I don’t have any data.”

This is not because we’re stupid or naïve; it’s that humanists have a very different way of engaging with evidence than most scientists or even social scientists. And we have different ways of knowing things than people in other fields. We can know something to be true without being able to point to a dataset, as it’s traditionally understood. We can know, to take just one example, that early silent film relied on the conventions of melodrama (http://www.worldcat.org/oclc/38501674) to create legible narratives, not because we have a spreadsheet somewhere, but because we’ve immersed ourselves so deeply in our source material that we’re attuned to its nuances.

That’s why humanists sometimes think you can make a visualization without data; because they want to illustrate ideas and movement, not necessarily data points as we’ve been discussing them here.

In fact, very few traditional humanists would call their source material “data.” You may have seen this piece (http://lareviewofbooks.org/essay/literature-is-not-data-against-digital-humanities) in the LA Review of Books in October 2012. While the language is pretty hyperbolic, I do think it helps to convey how uncongenial many humanists feel the notion of data is to the work that they actually do.

When you call something data, you imply that it exists in discrete, fungible units; that it is computationally tractable; that its meaningful qualities can be enumerated in a finite list; that someone else performing the same operations on the same data will come up with the same results.

This is not how humanists think of the material they work with.

This is not a perfect analogy, but imagine that someone called your family photograph album a dataset. It’s not inaccurate per se, but it suggests that this person just fundamentally doesn’t understand why you value this artifact. And it’s the same with humanists. With a source, like a film or a work of literature, you’re not extracting features in order to analyze them; you’re trying to dive into it, like a pool, and understand it from within.

Let’s take my silent film example again. It would be possible to enumerate all of the filmic conventions that recall the conventions of melodrama. Is there a villain? Is there a heroine? Are good and evil depicted in stark, black-and-white terms? You could even build a dataset like this and use it to show how film changed over time.
But, seriously, who cares? There's just such a drastic difference between the richness of the actual film and the data we're able to capture about it.

A dataset like this is so much less interesting than the trained judgment of someone who's seen many of these films and can turn a nuanced observation of these changes into a real argument.

(Of course, the video itself constitutes data, but I'll get to that in a second.)
And I would argue that the notion of reproducible research in the humanities just doesn't have much currency, the way it does in the sciences, because humanists tend to believe that the scholar's own subject position is inextricably linked to the scholarship she produces.

However. Things are changing, in ways both obvious and not. All of our stuff is on our computers now — all of it, from books to movies to archival documents. This is why, more than anything else, I think digital humanities is here to stay. If you *can* analyze something computationally, I think it's going to be really hard to tell people that they *shouldn't*.

This state of affairs has created some real problems for humanists, and, I would say, some real opportunities for libraries. If you speak to any historian who works in an archive, I *guarantee* you that they have hundreds, maybe even thousands of photos shot in an archive that look like this:

![Historical Dataset](http://miriamposner.com/blog/wp-content/uploads/2015/06/Screen-Shot-2015-06-25-at-1.49.06-PM.png)

Behold, the historical dataset.

This is *it!* This is how historians are organizing hundreds of archival photographs!! The best-organized among them are trying to manage these irreplaceable source documents in iPhoto. And incidentally, this is a big problem for me, as someone who works on the history of lobotomy.
It's not just historians who have a problem. Literature scholars, film scholars, everyone's dealing with lots of journal articles, video clips, and other sources, and are really struggling to organize them so that they can produce scholarship.

So humanists — even those who aren't digital humanists — desperately need some help managing their stuff, and libraries are in a great position to help them. I do feel that this is an underexplored opportunity space for libraries.

It's just that if you advertise that help as “data management,” they'll have no idea you're trying to talk to them.

I used to offer a workshop (http://miriamposner.com/blog/embarrassments-of-riches-managing-research-assets/) on “managing research assets,” and even that felt way too clinical to describe humanists’ sources. But if you get a chance to look at the blog post (http://miriamposner.com/blog/embarrassments-of-riches-managing-research-assets/) that contains all the suggestions I used in the workshop, you'll see that we're cobbling together dozens of tools, none of which really do what we want them to do.

So all of that is to say that even if they don't call their sources data, traditional humanists do have pretty pressing data-management needs. But the need becomes even greater when you're talking about people who consider themselves digital humanists — that is, people who use digital tools to explore humanities questions.

In many ways, digital humanists will have similar data-management needs to scientists and social scientists — they’ll have spreadsheets, images, and video, and will probably at least know what metadata is. In addition, the NEH Office of Digital Humanities, like the NSF and other funding agencies, now requires a data-management plan; so you will very soon encounter, I'm sure, a humanist approaching you at the 11th hour with a request that you write their data-management plan for them.

Just to give you a sense of the kinds of things humanists might do with structured data, I'll show you a project (http://dhbasecamp.humanities.ucla.edu/gettydata/) that my students (https://www.flickr.com/photos/101041253@N04/18843735855/) just completed, as part of a collaboration with the Getty Research Institute. The GRI maintains what seems to me really big data — about 1.5 million records relating to the transmission and sale of works of art (http://www.getty.edu/research/tools/provenance/), which they call the provenance index. It's a really complex and baffling database — a really great case study in humanities data, actually — because all of its records are derived from historical documents themselves, and so are eccentric, disparate, and historically and geographically uneven. But because it's so big, you can do interesting, unexpected things with it.

For example, one of my students got really interested not in the paintings but the frames themselves (http://dhbasecamp.humanities.ucla.edu/gettydata/frames/), which are fairly understudied within art history. He gathered sales data for paintings sold between 1689 and 1787, and through a combination of text analysis and secondary reading, determined that two major factors made a
frame valuable during this period: its beauty or its authenticity. With that information, he was able to show that there was indeed a market for frames described as “authentic” during this 100-year period.

So it's quantitative evidence that seems to show something, but it’s the scholar's knowledge of the surrounding debates and historiography that give this data any meaning. It requires a lot of interpretive work.

These are two relatively simple examples, but I think they do show a little bit about how digital humanists are tending to work with data.

First, we often find ourselves in conflict with publishers because of the kind of work we want to do. I mentioned that my IP address had gotten blocked, but this is mild compared to what has happened to other scholars; I definitely know people who've been threatened with lawsuits and the like for excessive downloading.

Second, we're not generally creating data through experimentation or observation — more often than not, we're mining data from historical documents. You name it, we've tried to mine it, from whaling logs to menus to telephone directories. This means that we tend to want different tools than scientists, and also that we have some interesting data-wrangling problems. More often than not, the categories that our historical sources used to divide up our data are not the same ones we're interested in analyzing. So we often have to do some very creative transformations and interpretation, as my student did with the frames data.

Third, it's just awful trying to find a humanities dataset. There are various humanities data repositories or registries, but they're terribly limited. And right now we're starting to see museums and cultural institutions releasing their data, and there's just no way to know who's released what, unless you're the kind of person who stays on top of these things. So we urgently need some help locating these datasets, aggregating them, and perhaps even linking them.

Fourth, we do need those web services Sayeed was describing yesterday, that are built on top of existing datasets. We are working a lot with APIs, and it's really insufficient for us to download one record at a time. And even for people who aren’t going to work with APIs, if you could build visualizations of datasets on the fly, or even just access the data in quantity, it would be a big help.

Fifth, we have a desperate need for help with data-modeling — and here is another place where I think libraries could really play a big role. This July, I'll be directing a summer institute on digital humanities and art history, and as I've been reading through the participants’ project ideas, I've been struck by how often it seems that what the scholar really needs is data-modeling advice. For example: the art historian who wants to show how and when art objects traveled across the Indian Ocean and relate that movement to corresponding changes in artistic practice.

What she really needs is a data model that can accommodate historical and artistic periods, geographic movement, and conventional time. Most humanities scholars are not trained to build these kinds of databases. But I think — I hope — that this is an area where the library could be a huge help.
Finally, we may even need new kinds of data specifications, because the currently existing standards for describing time and space, for example, are actually really inadequate for our needs. To give one example, many standards for specifying dates require time calculated down to the exact day, and sometimes even the minute or second. But humanists tend to deal in words like circa, spans of time, or things like “before” or “after” this event. Two technologists at Stanford actually took on this problem with a project called Topotime (http://dh.stanford.edu/topotime/). By specifying that certain characters represent things like uncertainty, contingency, or approximation, they've shown how we could move from depicting time as a point or a line to a much broader canvas of shapes.

Just as we need more nuanced data models for time, we find ourselves faced with a pretty limited palette of options for depicting important structures of power, like gender and race. Take the Union List of Artist Names (http://www.getty.edu/research/tools/vocabularies/ulan/), which is an incredibly important resource that places like museums use to establish authorities — that is, to make sure they're all using the same name to refer to an artist, and to associate that name with other data about the artist. It's a tremendously important resource, and without it museums couldn't share and network information; we'd never be able to figure out who holds what. But look how it deals with gender (http://www.getty.edu/research/tools/vocabularies/ulan/about.html)!

Now, the fact that it captures gender is crucial — otherwise we wouldn't be able to say that women are underrepresented in a museum's collection — but no self-respecting humanities scholar would ever get away with such a crude representation of gender in traditional work.

We find ourselves needing models for gender that can accommodate much more nuance than our current standards. For us, the proper mode of visualizing data may not be a pie chart; it may be a heat map.

So I don't know about you, but I actually find these problems to be quite interesting and challenging: taking the datasets we've been given — which were not at all created for our purposes — and working against their grain or reinventing them to try and tease out the things we think are really interesting.

It requires some real soul-searching about what we think data actually is and its relationship to reality itself; where is it completely inadequate, and what about the world can be broken into pieces and turned into structured data? I think that's why digital humanities is so challenging and fun, because you're always holding in your head this tension between the power of computation and the inadequacy of data to truly represent reality.

16 thoughts to “Humanities Data: A Necessary Contradiction”
2

Modelling

Learning outcomes of this chapter

• Putting linked data in a larger context
• Getting out of a hype-driven view on technology
• Understanding the importance of data modelling
• Making sense of data models and their serialization formats

1 Introduction

Metadata managers and the software they use often seem to have a striking resemblance with couples stuck in an unhappy relationship. During coffee breaks at conferences and workshops on metadata within the library and information science domain, it will not take you long to spot a circle of people engaged in what seems to be some form of group therapy. Do not be afraid. Go ahead and stand a bit closer. You will probably overhear typical phrases such as ‘We have been struggling to create new metadata fields for years!’ or ‘My XML export is terrible!’ Confronted with these laments, the group members will nod understandingly and express their sympathy. Complaining about one’s software is a popular point of discussion across the globe when collection holders come together to discuss metadata. Ironically, just like old couples who think twice about divorce due to the important emotional and economic consequences, metadata managers often persist for years in the abusive relationship with their software. They usually prefer not to move over to another software solution.

How different the ambiance in the digital humanities! Instead of complaining about the difficulties encountered with their database, people active in the digital humanities often are very proud about the information system they built to manage a specific type of resource or collection. Susan Hockey even coined the expression “Me and my database” papers. Anyone who has already attended a DH conference is familiar with the phenomenon: a researcher who presents, in tedious detail, how a database was developed to accommodate every peculiar
feature of a collection. These speakers tend to be very proud of the database they constructed and radiate love and passion for it.

Why do these two communities have such a different approach to the software they use to manage cultural heritage resources and their metadata? Why do collection holders constantly whine about their database, whereas digital humanists express their satisfaction and even brag about the happiness they found with their database?

These differences relate to the extent to which the model used to represent an object and its metadata is deemed adequate or not. When we want to make resources and their metadata available in a structured manner on the web, we first need to decide what characteristics of theirs are the most important to be represented. By doing so, we make an abstraction of the reality through the development of a model.

In the cultural heritage context we mentioned, institutions are forced to work with off-the-shelf software, since the development of a custom-built collection management system is simply not economically feasible. The drawback of working with existing software is that institutions often find themselves limited in how they can describe their objects. Vendors have a commercial incentive to develop generic software that can be sold to as many institutions as possible. This implies that collection management software already prescribes a certain explicit worldview, through the use of a pre-established model. It is therefore not always possible to accommodate the specific requirements of an institution and its collections, leading to frustration amongst collection holders.

In contrast, the DH community uses databases for limited and specific research projects, as they tend to focus on the documentation and publication of one specific type of resource or collection. Within these limited research projects with a precise scope, the requirements tend to be so specific that it is not possible to use off-the-shelf software. In this type of context, relational database management software (RDMS) is often used to implement a tailored model. The drawback of meeting all the precise requirements of such a project are the relatively high development costs and the difficulty to maintain the application over time. Investments are made in projects that very often cannot be re-used.

1.1 Deciding where to put the semantics

What does this have to do with linked data? The examples above demonstrate that both the use of a generic, standardized model and of a highly customized, specific model come at a cost. The tremendous amount of effort the LIS community has put into metadata standardization reflects how we have been trying to find a sweet spot between the two approaches. As it will be demonstrated through practical examples, the evolution from an unstructured
narrative to a highly structured representation of metadata requires the development of schemas in order to make the metadata interoperable. By slicing up unstructured descriptive narratives into well structured fields, we need to render the meaning of the different fields (also called attributes) explicit by documenting them in a *schema*. By structuring and atomizing metadata fields we make them more machine-interoperable, but we also become more and more reliant on the schemas when needing to interpret our own or someone else’s metadata. It is precisely in this context that linked data need to be understood. Through the adoption of a radically simple data model, abstraction can be made of the traditional XML and database schemas we had to use in the past to interpret and re-use data.

1.2 Getting away from a hype-driven view of technology

The adoption of a new technology is often illustrated in the form of the famous Gartner hype cycle (Lynden and Fenn, 2003). The graphical representation of the rise and decline of the popularity of a new technology draws attention to the exaggerated expectations which often accompany its introduction. After the so-called *peak of inflated expectations*, a technology tends to lose most of its appeal on the market a couple of years after its introduction. It is only after an extensive period that the technology reaches a stable level of adoption, based on its genuine added value in a production environment. One of the goals of this book is to teach you how to step away from a hype-driven view on technology by helping you understand not only the exact added value of linked data, but also its weak points.

Where should we situate linked data in this cycle? The recent enthusiasm to connect heterogeneous resources and to draw in new information from external knowledge sources perhaps recalls for some the unbounded enthusiasm the cultural heritage sector had for the eXtensible Markup Language (XML) around 2000 and a couple of years later for web 2.0. In hindsight, we can now safely say that both approaches have been (and continue to be) fundamentally important for how we create and manage our metadata. However, we should also acknowledge that neither XML nor the social web resolved all of the fundamental problems underlying how we can connect resources from various collections.

Despite a major overhaul of the general technological framework, illustrated by other developments, such as the maturing of open-source collection management systems and cloud-based hosting, for example, we are still very much facing the same problems the cultural heritage community was discussing almost five decades ago. For anyone working on the topic of digital cultural heritage, it is a humbling experience to read about the discussions that were taking place in the 1960s and 70s. In parallel with the creation of the Computer
Museum Network in 1967, a project was launched to create a common collection management database that would be used by all participants of the consortium. Numerous other initiatives have since been based on the same fallacy: if we all use the same tool, our metadata will become interoperable. Again and again, projects have demonstrated that even if people and institutions are using the same tools and standards, they implement them in different ways to accommodate the specific nature of their collections.

Are linked data here to break this vicious circle, or are we again confronted with an overhyped technology? Before we answer that question, we need to moderate the inflated expectations surrounding linked data. Practitioners trying to get to grips with linked data principles are frequently frustrated when confronted with the output of large-scale IT research projects. Huge volumes of metadata and controlled vocabularies have been converted over recent years into Resource Description Framework (RDF), producing billions of RDF statements. Unfortunately, these so-called triple stores only unlock their value through the use of a complex query language called SPARQL Protocol and RDF Query Language (SPARQL). The purely technology-driven nature of many linked data projects is leaving a bitter aftertaste amongst practitioners, who feel they need a PhD in semantic web technologies in order to take advantage of linked data.

1.3 The world’s shortest introduction to data modelling

Let us therefore, in this chapter, step away from the merely hype-driven view of linked data by choosing a more conceptual and historical approach. In order to grasp the potential but also the limits of linked data, we need a better understanding of the different data models which have been used over recent decades to manage metadata. The advantages of RDF, the data model underlying the linked data vision, can only be fully understood in the context of previous data models. At the end of this chapter you will understand that the different data models presented do not supplant one another, but continue to co-exist. The overview of the different models should make it clear that relational databases are here to stay, and will not be disposed of in favour of triple stores. Technology vendors and IT researchers have a tendency to overemphasize the role a new technology has to play. At the height of the popularity of XML, one sometimes got the impression that the back-end of any type of information system would become XML-based. A decade later, XML is criticized more often than not, and new serialization formats such as JSON are often preferred. This chapter will provide the world’s shortest introduction to IT fashion, in order to help you see the wood for the trees.

We will specifically focus within the overview offered in this chapter on the management of structured data, but be aware that the traditional barriers between structured and unstructured data are becoming increasingly blurred.
For decades, different communities have been working independently from one another on both topics. Database engineers focus on the optimization of the management of structured data, whereas computational linguists develop methods and tools to manage unstructured natural language in an automated manner. The different traditions and views between the two communities can be illustrated by analysing how both communities make use of XML. Computer engineers see XML as a hierarchical tree in which structured data can be encoded in order to facilitate the communication of data between machines. On the other hand, computational linguists and digital humanists look at XML as a method to insert small pieces of structure into an otherwise unstructured textual document. Indicating where exactly in a full text the names of places or people can be found allows scholars to automate to a certain extent the analysis of an unstructured corpus. We will discuss XML in more detail in the section on meta-markup languages later in the chapter. The traditional distinction between structured and unstructured data is particularly problematic in the context of metadata. For example, within a highly structured metadata record a descriptive field might occur which contains a narrative of several pages of unstructured full text. Should this metadata record be considered structured or unstructured?

The chapter will start with the most intuitive model for structuring data, which is tabular formats. Due to the limitations of this approach, the relational model was developed in the 1970s, remaining until today the standard to represent and manage complex data. As will be explained over the next sections, the appearance of the web towards the end of the 1990s catalysed the need for data portability. Sharing data between different databases is a very tedious process, for reasons which will be explained below. In order to facilitate the exchange of structured data on the web, meta-markup languages, and XML in particular, have been used from 2000 onwards. XML proposes a standardized syntax for the automated exchange of structured data, but the actual use and interpretation of the data can still be troublesome. The meaning of the elements and attributes of the XML files need to be defined in a schema. The interpretation of the schema remains a barrier for an automated consumption of data across information systems on the web. It is exactly here where RDF comes in. By adopting a data model which embodies the meaning of the data in its most essential and stripped down form, there no longer is a need for an outside schema to interpret and re-use the data.

Figure 2.1 compares the different models from a high-level perspective. You could consider this figure as a synthetic overview of Chapter 2. We are conscious that we are covering a lot of ground with this chapter. At times, it might be challenging to understand the interaction and links between the four different data models to be discussed. In order to help you put the individual sections of this chapter into a bigger perspective, Figure 2.1 highlights in an abstract manner the features of each data model. Even though each model has its own properties,
similarities have been highlighted insofar as possible. For example, rounded shapes represent individual data values; rectangular shapes indicate model-specific ways to add structure (with the exception of RDF, where arrows are used). Different shades indicate data values that semantically belong together, indicating how different models treat them.

1.4 Every advantage has its disadvantage (and vice versa)

Be aware that this chapter explicitly does not represent the different models as a linear succession of increasingly high-performing solutions to manage structured
data. In other words: please do not interpret the following sections of this chapter as a story of how we have gone from an inadequate approach towards the perfect solution. As with many things in life, advantages offered by a data model often imply disadvantages (and vice versa). For example, the schema-neutral feature of RDF comes at a big cost. Whether a data model (and the methods, technologies and tools it comes with) is suited for you entirely depends on the context of the problem you want to solve. To make things as clear as possible, every model will be illustrated with the help of small examples of metadata in relation to the work of Pablo Picasso.¹

1.5 Data models and their serialization formats

Before we proceed to the overview of data modelling, it is important to clearly distinguish each model conceptually from the formats that have been developed to serialize the data model. The serialization process converts data structures into a format. The format allows the translation of the information you are representing into a stream of bits that can be manipulated by software, communicated over a network, etc. The format allows a conversion from the bit level back into the original data.

The difference between a model and a serialization can be compared to a dish and its recipe. The in-memory model is the thing itself and is accessible for manipulation – or is immediately ready to be eaten, like a prepared dish. A serialization contains all necessary elements to construct the in-memory model, just as a recipe contains all the information you need to prepare the dish. Table 2.1 gives an overview of the different data models and their serialization formats which will be discussed over the next sections.

<table>
<thead>
<tr>
<th>Table 2.1 Data models and their serialization formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>data model</td>
</tr>
<tr>
<td>tabular data</td>
</tr>
<tr>
<td>relational model</td>
</tr>
<tr>
<td>meta-markup languages</td>
</tr>
<tr>
<td>RDF</td>
</tr>
</tbody>
</table>

2 Tabular data

Suppose you were given a collection of resources, such as photographs, books or DVDs, and were asked to describe the collection. What would be the most intuitive and natural thing to do? Chances are high that you would take a sheet of paper, or create a spreadsheet on your computer, and create columns in which you will aggregate the most important metadata of the resources, such as title, creator, date, etc.
2.1 Model

Conceptually, the world view you create with tabular data is comprised of columns and rows. The intersection of a column with a row gives meaning to the data contained in the particular cell. Figure 2.1 illustrates this data model on an abstract level. There is only one modelling dimension, consisting of the fields in the first header row. Each row contains data from different semantic entities, which we can also refer to as records. This is why tabular data are often referred to as flat files. Coming back to our concrete example, it is an intuitive act to use this model and to draw up a list as presented in Table 2.2, illustrating how you might develop a tabular overview of your resources.

Over centuries, catalogues and indexes were encoded in this tabulated form. The list, as most people would call tabular data, can probably be considered as the oldest information technology. Drawing up lists organized in columns is still often the first step taken when brainstorming and developing ideas about what metadata elements should be used to document a resource. Why is this such an intuitive data model? Tabular data offer the big advantage that they are almost self-explanatory. When reading a catalogue or an index in this format, you have a natural tendency to read in a horizontal manner by focusing on one line of the catalogue and reading from left to right the information gathered in the different ‘boxes’. This allows you to get an immediate overview of all the different metadata elements (in our table: title, creator and date) concerning one specific object. Semiologists or linguists would refer to the importance of the syntagmatic relations. Through the combination of different elements, meaning is created in the sense that we understand what an object is, when it was created and by whom. A vertical reading, gazing up and down the columns, allows you to get a sense of the different values of one specific metadata element. On this level, the so-called paradigmatic relations operate. These relations cluster members of the same category.

The difference and interaction between syntagmatic and paradigmatic relations might seem like a pedantic academy side note. Keep in mind that they play an important role in understanding the difference between the use of unstructured descriptions, which we can refer to as narratives, and structured metadata fields, which have been sliced up to make them more machine-processable. Lev Manovich drew attention to the fundamental difference between these two forms of presenting information:

| Table 2.2 Example of metadata encoded as tabular data |
|------------------|------------------|---------------|---------------|
| title            | creator          | date          | collection    |
| Guernica         | Pablo Picasso   | 1937          | Museo Reina Sofia |
| First Communion  | Picasso         | 1895          | Museo Picasso  |
| Puppy            | Koons, Jeff     | 1992          | Guggenheim    |
| ...              | ...             | ...           | ...           |

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
As a cultural form, a database represents the world as a list of items and it refuses to order this list. In contrast, a narrative creates a cause-and-effect trajectory of seemingly unordered items. Competing for the same territory of human culture, each claims an exclusive right to make meaning out of the world. The database (the paradigm) is given material existence, while the narrative (the syntagm) is dematerialized. Paradigm is privileged, syntagm is downplayed. Paradigm is real, syntagm is virtual.

Manovich, 2001, 231

Traditionally, people have privileged the form of narrative when communicating information, but the massive presence of database-driven applications on the web is reversing the situation. This evolution is very much embodied in how our metadata practices have evolved. From the beginning of the 20th century, our cultural heritage institutions have started to decompose the lengthy narrative descriptions drawn up by curators and transferred them to card catalogues and database records. This evolution drastically helped to facilitate search and retrieval, but actually making sense of a complex object is still based on the unstructured description. Manovich goes too far when presenting the two forms as exclusive and competing. It is the introduction of database-driven websites that made the advent of web 2.0 applications possible. As Chapter 5 will demonstrate, user comments can offer a valuable enrichment of the limited metadata an institution can provide, whereas named-entity recognition (NER) can currently be applied to facilitate complex search and retrieval procedures based on unstructured full text in natural language.

2.2 Serialization

The most popular serialization formats of tabular data are comma-separated values (CSV) and tab-separated values (TSV). The only, but important, difference between these two formats are the characters, appropriately called delimiters, used to indicate the separation between values. As their name indicates, CSV files use a comma as a delimiter, and TSV tabs. Please note that any type of character can be used as a delimiter. The CSV data from Table 2.2 are separated by a comma and rows are ended with a line break as follows:

```
title,creator,date,collection
Guernica,Pablo Picasso,1937,Museo Reina Sofia
First Communion,Picasso,1895,Museo Picasso
Puppy,"Koons, Jeff",1992,Guggenheim
```

The TSV version would look exactly the same, but the commas would be replaced by tab characters. And strictly speaking, the quotes around Koons, Jeff would

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
not be necessary because the comma has no special meaning. If, however, a value needs to contain an actual tab character, quotes would be necessary.

2.3 Search and retrieval
What are the implications of this data model for search and retrieval of metadata? A quick look at the metadata in Table 2.2 gives an overview of the limitations of the tabular file approach. For example, the name of the creator is expressed in different manners (‘Pablo Picasso’ and ‘Picasso’), as we encode the same reality every time when describing a new object this artist made. For a human being, it is straightforward to map these two different representations to the same reality. You have probably also noted the presence of “Koons, Jeff”, in which quotes are used to protect the comma separating the family and the first name.

When performing a full-text search on a string of characters, an algorithm will have more problems to deliver good search results. Let us therefore suppose that you want to update your metadata and encode the name of the creator in a uniform manner. Now imagine you do not have three records (as it is the case in our example) but a couple of hundred thousand . . . Managing your metadata in a tabular list would imply that you would need to go through all these records to see where one of the different spellings of the creator’s name appears and update it if needed to the preferred spelling. This working method is bound to introduce inconsistencies in your metadata. On a computational level, search and retrieval is very inefficient with this approach, as again the totality of your metadata records have to be checked to see whether they contain a specific value. Neither do tabular files offer the right tools to impose rules on how we encode values, resulting in inconsistencies in the way we encode metadata.

Problems only become worse when you start to think about searching across multiple files in this format. Another institution might have its own tabular data which contains relevant information for you, but how could you possibly perform a query across independent flat files in a consistent manner? Proficient users of Microsoft Excel could make use of macros and look-up tables to create links across multiple independent files, but these functionalities cannot be used outside Excel. This implies that you no longer have a platform- and application-independent format.

2.4 Change
How can tabular data evolve through time? The structure of catalogues and inventories does not change every month, but we could easily imagine at some point that we need to encode extra information, such as the technique (oil painting, aquarelle, etc.). Within the context of tabular data, we can simply add an extra column describing this new feature of the resources we are
documenting. If for some reason a column is no longer used or no longer contains relevant information, it can be deleted without any consequences for the rest of the data. Adding or deleting a column does not require you to make any modifications in the structure of the file. In this regard, an information system based on tabular data is resilient to change.

2.5 Implementation

Tabular data is one of the easiest conceptual formats, and as such, any software package will offer support. The most common form are spreadsheet applications such as Microsoft Excel, which in essence offer one giant table that can be modified. All spreadsheet software offers the possibility to export to TSV or CSV, albeit of course with the loss of formulas (cells that are calculated based on other cells), formatting (such as colour and borders) and functionalities such as macros, which we mentioned previously. Data types are also lost: all cell contents are stored as text.

Even with the simplest data model, a lot can go wrong in practice. Several elements are noteworthy here:

- Data can be separated by a comma but depending on a system’s local settings, this might actually be a semicolon! For instance, in many European languages, a comma instead of a dot is used as decimal separator in numbers (so 1,5 is actually 1½). On these systems, it would thus be impractical to use a comma as column separator, hence the choice of a semicolon – so CSV is not always true to its name. In practice, CSV has come to stand for any separator-delimited type, which confusingly also includes TSV.
- Rows end with a line break. Unfortunately, different systems can produce different results. For instance, on Windows systems, a line break actually consists of two characters (a carriage return followed by a newline), whereas on Linux-based systems, it is just a single character (only a newline). Additionally, Linux-based systems might expect the last line to end with a line break, while this is not necessary on Windows.
- There is no way to indicate the difference between the header row and the rest of the data. This means that we will have to tell this explicitly to the parser.
- If the field value itself contains a comma or a line break, such as Koons, Jeff here, the value is typically enclosed in double quotes, so it can be parsed correctly as a single value and not as multiple rows or columns. Note that not all parsers support both cases; line breaks, especially, might be confusing. In general, enclosing any field with double quotes is allowed, even if no special characters occur within the value.
- Another dangerous issue is character encoding. Different systems use
different byte codes to represent characters, in particular if these characters lie outside the traditional ASCII alphabet, such as accented letters or Japanese characters. If one system has written a file in a certain encoding, it is important for another system to use the same encoding to read the file. Otherwise, an accented character in one encoding might accidentally be transformed into one or more other characters in a different encoding. This phenomenon is called *mojibake*, the incorrect presentation of characters due to an encoding mismatch. Chapter 3 will explore how the above-mentioned issues impact metadata quality and what can be done to mitigate them.

- What if the field value contains a double quote? This is solved by *escaping* the quote, adding a character in front of it that signals the next character has no special meaning. In the case of CSV, this escaping is done by doubling the quote. For instance, the value `width: 7", height: 5"` is encoded as "width: 7"", height: 5"", wherein each literal quote is preceded by another.

- The main problem with CSV is that there are many ways to *encode* a file. The Internet Engineering Task Force (IETF) proposed a standard way of serializing CSV (Shafranovich, 2005), and this format can be read by most parsers. However, this by no means implies that all generators will follow this standard. Fortunately, most parsers are *adaptive*: they apply a heuristic on the file in question to determine which conventions were used. For instance, if every line in the file contains an equal number of semicolons, it is likely to assume that the delimiter is a semicolon. Also, if the third column always consists of decimal digits, except the first row (as in our example), then it can be assumed that the first line contains header data. Of course, none of these strategies are perfect; in practice, human verification is necessary for correct parsing.

3 Relational model

The relational model was developed to deal with the issues related to redundancies and inconsistencies as described above. Developed at the end of the 1970s, the relational model has been by far the most successful approach for managing structured data, and will continue to be used in the decades to come.

3.1 Model

The model asks you to take a step back from the individual metadata recorded in the tabular format and to identify on a higher level what the different *entities* are in the reality that you want to represent. We may define an entity as the ‘type of information that varies independently of another’ (Ramsay, 2004). Each
entity is characterized through the use of attributes. As depicted in Figure 2.1, entities are connected through relations to one another. Every record contains a unique key per table, which other records can use to refer to it in a relation.

3.2 Design methodology

Building a relational model is a difficult task that requires a lot of experience. Nonetheless, there are some guiding principles that you can use:

- The first task is to discover the entity types that the database will contain. Typically, entities correspond to independent concepts in the world of which there will be many, and each one has properties of its own. In our example, it is certain that ‘Work’ will be an entity type, as the database will contain several works. It is also likely that ‘Creator’ will be an entity type, as it is independent from Work and there will be many of them. However, an entity type of ‘Country’ will probably not be necessary, as we will only need the country’s name and no other properties. However, for other use cases, it might be meaningful to encode ‘Country’ as an entity type.

- For every entity type, a table will be created in the database. Each row in the table will have a unique identifier, often a numeric value that is automatically generated. Each property of the entity will be a column in the table, and each column can have a value type. Our ‘Work’ table might have a textual field for the title and a date field for the creation date (or a four-digit field if we only plan to store the year).

- Next, one-to-many relationships must be modelled. They provide a mechanism for a record in a table to link to one record in another table, and the record in the second table can receive several such incoming links. As each entity has a unique identifier, we can add a column to the first table that will contain this identifier. That way, the objects in the first table can point to an item in the second table. For instance, the ‘Work’ table should contain a ‘Creator’ column that stores the identifier of the creator. That way, each work can be associated with one creator, and each creator can then be associated with several works.

- Finally, many-to-many relationships should be described. As fields in databases are traditionally not multi-valued, we must find another way for a record in a table to point to several records in another table. This is done by having a third relationship table, which has one column for identifiers of the first table and one column for identifiers of the second table. That way, we can find all items that belong together by traversing the rows of this table. Additional columns might describe properties of the relationship. In case works are authored by many creators, we could opt to represent them as a
many-to-many relationship. Then, we would have a third table called ‘Creatorship’ with columns ‘work’ and ‘creator’ that store the respective identifiers. We might add a textual column ‘role’ describing how the person was involved in the creation of the work.

The above points are merely guidelines. In practice, there are many possible motivations behind the choice for a certain design decision. There will always be trade-offs between optimal modelling flexibility, performance and simplicity. This is well illustrated by the option of whether to allow multiple creators for a work. This might allow you to describe the reality more closely, at the expense of a more complex (and possibly slower) data model.

Let us come back to our example. In order to have a sufficiently complex scheme, extra attributes, such as style, were added, as in Figure 2.2. It should be clear that there is no such thing as one unique and perfectly adequate model, as the same reality can be interpreted in multiple ways.

Depending on the importance you give to an aspect of the reality you are modelling, you either decide to consider it as an entity or as an attribute. Despite the simplicity of the scheme in Figure 2.2, many readers of this book probably would come up with different versions of the scheme. For example, one could decide to reduce the entity Collection to an attribute of the entity Work, if you do not consider the address as an independent aspect that needs to be

---

**Figure 2.2** An example entity relation model with the relationships highlighted

---

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
documented. The process of placing separate entities in separate tables is called *normalization*. Unfortunately, the more tables that need to be accessed to reconstruct the related metadata of an item, the slower operations will become. Therefore, a meaningful balance should be found between what data will be stored in a single table and what data is stored as a relationship.

The added complexity on the modelling level is made up for by the advantages offered by having a single record for an entity, that can be referred to with a unique ID. For example, every time you need to refer to the fact that an object is housed in a specific collection, you do not have to re-encode the metadata in relation to the address of where the collection is managed and other attributes of the entity Collection. You simply refer to the ID of the collection, and the same applies to other entities such as Creator. This approach ensures a lot more consistency. Those IDs are typically *indexed* to ensure the corresponding rows can be fetched in a fast way without traversing the entire table.

### 3.3 Implementation

Software built on top of this model, referred to as *relational database management software* (RDMS), has been extensively developed over the last decades and is currently at a very mature stage. Everyone has probably heard at some point about MySQL, the most popular open-source RDMS used for web applications, or MS-SQL, a proprietary RDMS developed by Microsoft. Another well known manufacturer is Oracle. Collection management systems, archival inventories and library catalogues are all built on top of a RDMS.

Most database systems work in a client-server set-up, where the server runs a RDMS and the client interacts with it using SQL statements. Consumer and small business applications, such as Microsoft Access, simplify the structure by offering a graphical user interface that works directly on top of a local database file. For regular RDMS, graphical interfaces for clients exist as well, but the communication underneath is done in SQL.

For instance, a table in MySQL can be created with:

```sql
CREATE TABLE Work (  
id INT AUTO_INCREMENT PRIMARY KEY,  
title VARCHAR(100),  
creator INT,  
collection INT,  
year CHAR(4),  
style VARCHAR(40) );
```

This makes a new table called *Work* with an integer *id* column (*INT*), a textual
The title column (VARCHAR(100)), meaning a string of characters with variable length, maximum 100), and integer creator and collection columns a 4-character year column, and a 40-character style column. The id column is special, since it should provide a unique identifier for each record. Therefore, it has the labels AUTO_INCREMENT (so new numbers are assigned automatically) and PRIMARY KEY (so the database knows that this is a unique field).

To insert data in this table, we can use:

```
INSERT INTO Work (title, creator, collection, year, style)
VALUES ('Guernica', 43, 20, 1937, 'Cubism');
```

We supply the table name, followed by the names of the fields and then the values for these fields. Strings are surrounded by single quotes (and single quotes within strings are escaped by a backslash). Note how we did not supply a value for the id field, as this value is automatically generated (and will default to 1, 2, 3, ... on an empty table).

### 3.4 Search and retrieval

After several records have been inserted, we can retrieve them with SELECT queries. For instance:

```
SELECT * FROM Work;
```

will select all records in the Work table. If we only want the titles of works by Picasso (assuming he has identifier 43 in the Creator table), we can do:

```
SELECT title FROM Work WHERE creator=43;
```

This is only a basic introduction to SQL, as end-users are only confronted with predefined SQL queries accessible through a graphical interface of a collection management system. However, it is important to understand through the example the logic behind the SQL query language. Section 5 will build further on this example to illustrate how the SPARQL query language works.

### 3.5 Change

The previous section on the tabular format described how little impact change has on the structure of a flat file. Adding new columns or deleting existing ones does not fundamentally alter how the tabular data can be used. The situation could not be more different with relational databases. Adding an extra table requires the database manager to rethink the entire schema of the database, as...
adding an extra table might imply a degrading of the normalization process.

Let us come back to our example. Imagine we want to add an entity \textit{ArchivalItem} which describes the archival holdings of an artist that the institution possesses, such as correspondence, notes, personal photographs, historical press clippings, etc. How do we update our database with minimal effort? We can create a table \textit{ArchivalItem} with the attributes \textit{ID, document type, year, creator,} and \textit{collection}. Then the change can happen by just adding this single table. However, that table would carry a considerable overlap with \textit{Work}, as both have a creator, collection, and year. This information spread becomes difficult to manage in the end. If we want to respect the normalization requirements, we cannot just add extra tables, but we also need to modify the existing tables. Figure 2.3 shows a better integrated solution: the common attributes of \textit{Work} and \textit{ArchivalItem} are placed into a separate \textit{Asset} table. However, this means that the existing table and data structure has to be modified. Apart from ensuring the normalization of the new database schema, the

\begin{figure}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
ID & creator & collection & year \\
\hline
5 & 43 & 20 & 1937 \\
16 & 57 & 18 & 1992 \\
32 & 43 & 20 & 1948 \\
\hline
\end{tabular}
\caption{To support archival items in a consistent way, the \textit{creator, collection,} and \textit{year} fields must move from the \textit{Work} table into a shared \textit{Asset} table}
\end{figure}

modifications also impact external systems, such as the public front-end built to give access to the data on the web. Performing these types of updates and modifications every couple of months can be very cumbersome. In practice, these modifications are often avoided, as there is no time to fundamentally rethink the structure of the database. In this context, people often rely on lightweight and ad hoc solutions, such as creating a standalone spreadsheet. This type of short-term decision causes, over a period of years, tremendous issues with data consistency, as reference data are scattered across different applications. We can

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
therefore conclude that it is not a trivial matter to update and maintain a database, due to the complexity of modifying the database schema.

3.6 Sharing
As referred to in the beginning of this chapter, institutions already thought about interoperability between collections right from the start when RDMS were implemented in some pioneering cultural heritage institutions. There was (and there still is) a strong belief that acquiring the same collection management system provides the needed basis for interoperability. However, the customization of these software tools to accommodate specific requirements of each institution more often than not resulted in different approaches regarding the use of metadata elements. This made the exchange of records between institutions, which might have been using exactly the same software, problematic. Letting databases talk to one another and share their content is a complex matter, regardless of the application domain.

At this stage, it is important to point out the difference between binary and non-binary files. The previous section illustrated how tabular file formats make use of text files, allowing you to open a .tsv or .csv file with any standard text editor or spreadsheet software, making the exchange of metadata very straightforward. Databases, however, are stored in binary files which introduce a dependency on a specific software application. If you wanted for example to re-use a database of an institution, you would be obliged to use the same RDMS. Licences for proprietary RDMS easily cost around US$10,000 and you could potentially run into compatibility problems if you used different versions of the same software. You could also create a Structured Query Language (SQL) query, allowing you to create a data dump, and to import it afterwards in another application. But even if a standardized version of SQL exists, be aware that vendors implement the standard in varying ways. Certain RDMS have their own proprietary extensions, for example for column types, leading again to potential data compatibility problems.

We can therefore conclude that the interoperability of databases is quite problematic. Fortunately, methods have been developed to facilitate the export and import of structured data from and into different databases.

4 Meta-markup languages
Before we get into the details of how XML is used to facilitate the exchange of structured data, this section will make quite an extensive detour to the origins of markup and meta-markup languages. XML is probably the most abused and incorrectly used acronym (apart from RDF) at meetings in the cultural heritage sector. Some people consider it a programming language, others think it will automatically make their metadata smart and semantic. A broader view on the
origins of XML will allow you to understand the tremendously important difference between applying *markup* and *makeup*. Understanding the difference between a *data-* and a *narrative-centric* view of XML will also allow you to better understand why tool or standard A is better than tool or standard B, depending on whether you are managing text or data. Moreover, the evolution of XML is very much intertwined with the development and the future of HTML. The relevance of initiatives such as Schema.org or the OpenGraph protocol will also be better apprehended with a good understanding of the global context of meta-markup languages.

### 4.1 Adding structure to content

In parallel with the work on the development of relational databases for the management of structured data, producers of large volumes of unstructured texts, such as the pharmaceutical or aeronautics industry, developed the concept of a meta-markup language throughout the 1970s and 1980s. These industries are confronted with the need to manage complex and voluminous documentation of production and safety guidelines. In order to streamline the typesetting of these complex text documents, the idea was developed to make use of *markup* to indicate the presence of *structural elements* (title, subtitle, paragraph, etc.) inside a document.

Markup can be thought of as *annotations* added to a document. A manuscript would be annotated with signs indicating how specific parts of the text should be displayed. Through the use of *delimiters* (remember the role these play within the tabular model), such as angle (⟨⟩) or square brackets ([ ]), the markup is clearly distinguished from the text itself. The characters used to indicate the markup are purely a matter of convention, one could also use other characters such as $ or *.

For example, if a specific string of characters which represents the title of a section should be printed in a large bold font, you could have the following HTML markup:

```html
<font size="6"><b>Introduction to metadata</b></font>
```

Your browser would then render the text as:

**Introduction to metadata**

The above example is a typical illustration of how *markup* is merely used as *makeup*. The markup simply indicates how one particular string of characters, in our example ‘Introduction to metadata’, should be presented. Imagine you have a document containing several hundreds of section titles. Instead of
presenting them in size 20, you actually prefer to have them slightly bigger. Making this modification with the above approach obliges you to manually update the markup for every section title in your document.

Conceptually, a whole new world of opportunities for automated processing appears when the markup focuses on the function of a specific string of characters within the structure of a document. Instead of hardwiring how every individual element of a text should be presented, the markup can indicate the role it plays within a text. Let us re-use the same example and apply this time genuine markup and not makeup:

```html
<h1>Introduction to metadata</h1>
```

We no longer indicate how the string of characters ‘Introduction to metadata’ should be printed. Instead, we specify the role this string plays within the text, by stipulating that ‘Introduction to metadata’ is the title of a section. You can re-use the markup element `h1` for all the titles of sections within the document. You only specify once how this specific structural element of your text should be formatted. This specification can take place either in the header of the document file or in a separate file linked to your document, which could contain the following definition:

```css
h1 {
  font-size: 20pt;
  font-weight: bold;
}
```

This gives you the tremendous advantage that you can define in one central place how a specific structural element within your document should be displayed, from where it will then be implemented in a coherent manner across the entire document. Once the definition of the layout of a document is contained within a separate file, one can imagine having multiple files linked to a document in order to automatically switch between different designs. This is the idea behind the ‘write once, publish many’ principle. In a web context, it lets you automatically switch between a website with bigger or smaller fonts, or a standard version of a web page packed with images and colours and a Spartan page optimized for printing in black and white. The content stays the same, you just use another style sheet which indicates how the different elements of the web page should be rendered.

4.2 Model

Now that you understand the purpose of markup, we need to conceptualize its
use. In order to make markup machine-processable, you cannot just randomly put commands contained within delimiters inside a document. The markup needs to respect a logical and consistent structure to be processed automatically.

Conceptually, you can think of marked-up documents as trees. They have one root and consist of branches which themselves contain smaller branches, as depicted in Figure 2.1. A node and its directly descending nodes have a parent–child relationship; all directly descending nodes of a parent are siblings. The hierarchical nature of this data model is central: child elements inherit by default all of the characteristics which have been predefined on the level of the parent element. However, this default inheritance can be overridden if a specific characteristic is defined on the level of the child element.

4.3 Meta-markup
Why have we called this section ‘meta-’ markup languages? The model described above asks you to define a hierarchy of structural elements which you could consider as the building blocks of the documents you manage. Thinking about books, one could easily say that the element ‘book’ represents the root level of the document, which encloses all other elements. A child element of the root would be ‘chapter’, which itself consists of a title and multiple sections. We could imagine that this standardized vision of a book could be re-used by many people. But perhaps you like to use an epigraph at the beginning of every chapter, whereas other people would never make use of this element.

The early developers of meta-markup in the 1960s and 1970s foresaw that different application domains would have very different needs for the structural elements of their documents. For example, stanzas are an important structural building block of classical poetry. The documentation of production phases and testing of drugs might have specific elements which structure the quality procedures that need to be respected during the development of a drug.

It was therefore decided not to predetermine all potentially interesting markup elements. Instead, a syntax and grammar was developed which allows everyone to develop their own specific markup language. Hence the use of meta, which refers to something at a higher, more abstract level. Out of the idea of a meta-markup language, the Standard Generalized Markup Language (SGML) was born. In hindsight, the heritage and impact of SGML, adopted as an ISO standard in 1986, has been enormous. SGML was used as a conceptual foundation for all the major standards which made the web a success: HTML, XML and CSS. Ironically, SGML has been considered by most as a failure, as the industry never largely adopted the standard due to its complexity. Bob Boiko’s metaphor ‘SGML became like those backwoods blues players of old to whom the pop stars give honor but no money’ sums up the situation quite correctly (Boiko, 2005). The use of SGML required a thorough analysis of the
domain and content to be represented, followed by an intensive modelling exercise in which all essential structural elements of a document had to be predefined in a schema. The implementation itself was terribly expensive, due to costly software and the need for highly specialized staff. The success of HTML can be linked to exactly the opposite conditions: anyone can write HTML and the tools are freely available.

4.4 The Hypertext Markup Language

Around 1990, Tim Berners-Lee developed and implemented HTML. SGML was a major source of inspiration, but for reasons of simplicity a fixed set of elements was defined, representing the basic building blocks of a web page (e.g. `<head>`, `<title>`, `<body>`, `<link>`). Notice how these elements indicate structural elements of a web document, and do not stipulate any layout. This markup is to be parsed by a web browser, which is responsible for interpreting the HTML tags and for displaying the web document on the computer screen. HTML is therefore a markup language (and not a meta-markup language). This implies that you can only make use of a pre-defined set of tags which can be interpreted by a web browser. Nothing holds you back from inventing your own HTML tags, but in order to use them you would need to build your own browser.

Needless to say, HTML was a success. However, after a decade Berners-Lee’s brainchild was corrupted from a markup into a makeup language. The focus on the aesthetics (and not on the semantics or structure) of web pages was beginning to undermine the potential of the web as a global information system. What happened?

From the mid to the end of the 1990s, web publishers were building up the dot com bubble. During this period, one of the biggest business model underlying the web was born. Within this model, the value of a company does not lie in its net income acquired through the commercialization of a product or a service offered to its user base. The mission of a company is to rapidly build up a user base by offering a free commodity (email, social networking, photo sharing, etc.). This business model is based on the assumption that the company will be able to monetize its customer base at a latter stage through advertising and the aggregation of consumer profiles for example.

With this in mind, it is easy to understand why web developers focused in the first place on an attractive and distinctive layout. This tendency played an important role in the browser war between Netscape Navigator (precursor of Mozilla Firefox) and Microsoft Internet Explorer. In order to attract the biggest user base, both browsers developed, independently one from the other, HTML elements that would render web content attractive and original. To fully understand the impact of these practices, do the following small exercise. Launch Notepad or any other simple text editor and encode the following HTML:

```
32 LINKED DATA FOR LIBRARIES, ARCHIVES AND MUSEUMS

domain and content to be represented, followed by an intensive modelling exercise in which all essential structural elements of a document had to be predefined in a schema. The implementation itself was terribly expensive, due to costly software and the need for highly specialized staff. The success of HTML can be linked to exactly the opposite conditions: anyone can write HTML and the tools are freely available.

4.4 The Hypertext Markup Language

Around 1990, Tim Berners-Lee developed and implemented HTML. SGML was a major source of inspiration, but for reasons of simplicity a fixed set of elements was defined, representing the basic building blocks of a web page (e.g. `<head>`, `<title>`, `<body>`, `<link>`). Notice how these elements indicate structural elements of a web document, and do not stipulate any layout. This markup is to be parsed by a web browser, which is responsible for interpreting the HTML tags and for displaying the web document on the computer screen. HTML is therefore a markup language (and not a meta-markup language). This implies that you can only make use of a pre-defined set of tags which can be interpreted by a web browser. Nothing holds you back from inventing your own HTML tags, but in order to use them you would need to build your own browser.

Needless to say, HTML was a success. However, after a decade Berners-Lee’s brainchild was corrupted from a markup into a makeup language. The focus on the aesthetics (and not on the semantics or structure) of web pages was beginning to undermine the potential of the web as a global information system. What happened?

From the mid to the end of the 1990s, web publishers were building up the dot com bubble. During this period, one of the biggest business model underlying the web was born. Within this model, the value of a company does not lie in its net income acquired through the commercialization of a product or a service offered to its user base. The mission of a company is to rapidly build up a user base by offering a free commodity (email, social networking, photo sharing, etc.). This business model is based on the assumption that the company will be able to monetize its customer base at a latter stage through advertising and the aggregation of consumer profiles for example.

With this in mind, it is easy to understand why web developers focused in the first place on an attractive and distinctive layout. This tendency played an important role in the browser war between Netscape Navigator (precursor of Mozilla Firefox) and Microsoft Internet Explorer. In order to attract the biggest user base, both browsers developed, independently one from the other, HTML elements that would render web content attractive and original. To fully understand the impact of these practices, do the following small exercise. Launch Notepad or any other simple text editor and encode the following HTML:

```
```
<html>
<blink>This text only blinks in Firefox</blink>
</html>

Make sure to save the document with the .html extension. Now start up the Firefox web browser and open the HTML document you just created. Congratulations, you have just created your first makeup’d web page: the text contained within the <blink></blink> tags should blink. Now open up the same document in any other web browser (Microsoft Internet Explorer, Google Chrome, Apple Safari...). Nothing happens. The browser understood that there are tags but does not understand them and therefore just displays the text contained within them.

The blink element is a non-standard presentational HTML element introduced in Netscape Navigator, but not supported by other browsers. Anyone who is old enough to have surfed the web in the late 1990s will think fondly of all the weird and utterly user-unfriendly websites which held flashing and hovering content. On top of that: exactly the same page displayed differently across browsers.

4.5 The eXtensible Markup Language (XML)

The interoperability issues described above, coupled with the exploding volume of HTML documents containing no exploitable structure or semantics, resulted in a growing unease within the information retrieval and information science community. A standard was needed to ensure a more structured web, and XML saw the light. The standard is built as an application profile of SGML, but simplifies its use. An effort was made to keep 80% of SGML's functionality with only 20% of its complexity.

XML being a meta-markup language, realizes that user communities have the possibility of defining their own markup elements, hence the adjective 'extensible' in the name of XML. The big advantage of XML, especially at the time of major incompatibility issues on the web, is its platform and application independence. With its open and standardized format, XML allowed the web community to make a big step forward with the publication of structured content.

4.6 Designing XML documents

Similarly to relational database schemas, the design of XML documents also involves extensibility and simplicity trade-offs. The main discussion in XML is whether to model an entity as an element (serialized as tags surrounded by angle brackets) or as an attribute (key/value modifiers of a tag). Every XML document begins with an XML declaration, a processing instruction that identifies the document as a specific XML version. Processing instructions are special tags that
start and end with question marks inside the angle brackets. For instance, a minimal XML document for our collection would be:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Art title="Modern art"/>
```

So what we see here is the XML declaration, followed by an `Art` tag that has a title attribute with value `Modern art`. The `Art` element is the root element of our document, and every XML document should have exactly one root element. Since there are no other elements yet, we have made `Art` self-closing by including a slash before its ending angle bracket. Note how we were free to choose the names of the tag and the attribute, in contrast to more specified languages such as HTML. That does however not imply total freedom: the mandatory version attribute and its value are predetermined by the XML standard. The encoding attribute is not mandatory, but as we said before when explaining tabular data, plain text files always have a risk of being interpreted in a different encoding from that intended. Therefore, by specifying the encoding, we ensure that the interpretation will happen uniformly.

The root element is not difficult to get right, but modelling questions arise when we add data elements. For instance, let’s add a work to the collection.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Art title="Modern art">
  <Work title="Guernica" year="1937" creator="Pablo Picasso" collection="Museo Reina Sofia" location="Madrid"/>
</Art>
```

The hierarchical structure of XML documents now becomes apparent: the `Work` element is a child of the `Art` element. Initially, we have chosen to model the work’s elements as properties. However, this might not prove extensible enough. For instance, it is difficult to add more structure to the creator field, and there is currently no relation between the collection and location fields. The opposite approach would be to model everything as child elements:

```xml
<Art title="Modern art">
  <Work>
    <Title>Guernica</Title>
    <CreationDate>
      <Year>1937</Year>
    </CreationDate>
    <Creator>
      <FirstName>Pablo</FirstName>
    </Creator>
  </Work>
</Art>
```

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
This leaves us maximum flexibility to extend the document at any point. However, this also comes at a cost: the hierarchy is now relatively deep to express simple concepts, even for straightforward properties such as a year of creation. Even though the original design goals for XML state that ‘terseness in XML markup is of minimal importance’, it might be important for our application. Although software does not have any more difficulty parsing hierarchies as opposed to attributes, unnecessary complexity is never an asset. Understanding the XML document at a glance becomes more difficult for humans (and XML was designed to be read by both humans and machines), and the job of programming the format reader on top of the XML parser becomes more complex. In practice, a compromise often works best:

Here, we have chosen to model all values that will not be decomposed or require further properties as attributes. For instance, a work’s title does not require further description, but we would add additional information to a creator, such as date and place of birth.

This might remind you of the discussion on relational model design, where we first determined entity types. Indeed, the decision is similar: things that would end up as entities in databases are likely represented as elements in an XML document as well. In contrast with databases, XML documents are more flexible and there is an even larger grey area for modelling choices. As always, this extended flexibility comes at a cost: databases are made for rapid data search and manipulation; searching XML documents is more than an order of magnitude slower.

Speaking of databases, you might wonder how to represent relations in XML. The answer is that you are free to choose that, but some choices are wiser than others.
For instance, we can simply continue the model as above and add another work:

```xml
<Art title="Modern art">
  <Work title="Guernica" year="1937">
    <Creator firstName="Pablo" lastName="Picasso"/>
    <Collection name="Museo Reina Sofia" location="Madrid"/>
  </Work>
  <Work title="First Communion" year="1895">
    <Creator firstName="Pablo" lastName="Picaso"/>
    <Collection name="Museo Picasso" location="Barcelona"/>
  </Work>
</Art>
```

However, this duplication of information is harder to maintain and it might lead to errors. In fact, the last name of the creator of the second work is incorrectly spelled 'Picaso', even though it appears correctly in the first. Therefore, it makes sense to model the information only once and refer to it using identifiers:

```xml
<Art title="Modern art">
  <Work title="Guernica" year="1937" collectionId="Co20">
    <CreatorRef creatorId="Cr43"/>
  </Work>
  <Work title="First Communion" year="1895" collectionId="Co22">
    <CreatorRef creatorId="Cr43"/>
  </Work>
</Art>
```

In contrast to database systems, you are responsible yourself for the correct assignment and use of identifiers. Note how we modelled collectionId as an attribute of work, but Creator as a child element. The rationale behind that is that a work only resides in one collection, whereas there might be many creators of a single work, and a separate element allows us to specify a role for each of them (as with the many-to-many relationship of a database schema). This design choice allows the specification of multiple creators, since attribute names on an element must be unique.

### 4.7 XML Schema

The flexibility of XML documents might seem a drawback if you want to
consume XML. After all, your application will expect to see specific elements and attributes, but if anybody has the freedom to create their own, how can you be sure that the things you need will be there? We briefly mentioned before that this is possible with an XML schema, a document that explains what kind of XML markup is allowed.

Different languages exist to express schemas, the oldest being Document Type Definition (DTD), part of the original XML specification. The DTD specification of our Work element might look like this:

```xml
<!ELEMENT Work(CreatorRef)+>
<!ATTLIST Work title CDATA #REQUIRED>
<!ATTLIST Work year CDATA #REQUIRED>
<!ATTLIST Work collectionId IDREF #REQUIRED>
```

We again note a special kind of tag, which starts with an exclamation mark. The above fragment states that Work is an element that can contain many CreatorRef elements. It can have title and year attributes of type CDATA (character data) and a collectionId attribute that is an IDREF (a reference to an identifier), all of which are REQUIRED. This allows a parser to check whether the Work element is specified in the right way. Additionally, it can verify whether the identifiers are used correctly, as it will check for each IDREF attribute whether an element with this ID exists.

However, DTD has a quite peculiar syntax and it does not have a strong expressive power. For instance, we could not specify that year is a numeric value. Also, more complicated hierarchical rules cannot be efficiently described. Therefore, XML Schema (note the capital ‘S’) has been created by W3C (the World Wide Web Consortium). It features an XML syntax to describe schema documents, which themselves can also be validated by XML Schema. A description of the Work element would be:

```xml
<xsd:element name="Work">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="CreatorRef" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="title" type="xsd:string"/>
    <xsd:attribute name="year" type="xsd:gYear"/>
    <xsd:attribute name="collectionId" type="xsd:IDREF"/>
  </xsd:complexType>
</xsd:element>
```

This says that Work is an element that can have several CreatorRef elements. It
can have a title string as attribute, a year that has a year data type, and collectionId which is an IDREF. We see that the XML Schema syntax is more verbose, but it is also more expressive. For instance, the year field is now specified more precisely thanks to XML Schema built-in data types.

For documents with an associated DTD or XML Schema, various automated validators exist that either guarantee the validity of a document or show what type of errors occur. Many software libraries for XML parsing support this functionality. Checking the validity of an XML document upfront means the rest of your software chain can read and manipulate the document as expected, without causing errors because of missing or incorrect structure.

4.8 Namespaces

As anyone can make their own elements and attributes in an XML document, we need a mechanism to universally identify which ones are the same. For instance, two documents might use a title element, but one uses it to designate book titles, and the other for personal titles such as Mr or Mrs. While enforcing a specific document structure, schema documents alone do not provide a means for consistent re-use across different types of documents that need to re-use the same elements in another context.

This is the issue that XML namespaces address – they are a method of qualifying element and attribute names (Bray et al., 2006). Namespaces allow you to re-use what has already been developed by someone else, and by doing so you can explicitly state that you agree with outside parties on how your data should be interpreted. The link with metadata schemes is self-evident here, as they share the same goal: making explicit statements about how a specific value should be interpreted. For example, if I want to use an element in my XML document which represents the name of a creator, it would make a lot of sense not to issue an identifier on my own for that element, but to re-use the namespace issued for the Dublin Core element “Creator”: http://purl.org/dc/terms. When used on a creator element, it indicates that this element is to be interpreted as defined by the Dublin Core schema, which defines it as “an entity primarily responsible for making the resource”.

Namespaces can be indicated using the reserved XML attribute xmlns on an element, which then holds for this element and all of its descendants. Namespace declarations are mostly seen in the schema element, forming the root of the schema, and are applied to the entire document. For instance, an XML document could start with the following tag:

<Agent xmlns="http://purl.org/dc/terms/"

This indicates that all elements in the document, including the root element, are
to be interpreted according to the Dublin Core specification, which defines this namespace. Multiple namespaces can be used in a single document by using prefixes. For instance, we could re-use elements from a generic schema such as Dublin Core in combination with more specific elements from VRA Core, as follows:

```xml
<Art title="Modern art"
     xmlns:dc="http://purl.org/dc/terms/
     xmlns:vra="http://vraweb.org/vracore4.htm">
  <Work>
    <dc:creator>Pablo Picasso</dc:creator>
    <dc:title>Guernica</dc:title>
    <vra:technique>Oil painting</vra:technique>
  </Work>
</Art>
```

### 4.9 Search and retrieval

While relational databases are especially designed for maximal performance, XML documents are designed for maximal flexibility. As we have seen, information can be modelled in different ways. As such, we cannot expect XML to achieve the same level of speed for search and retrieval, even if the entire model is loaded into memory (which is not always possible, due to size constraints). Nonetheless, like databases that are accessible through SQL, XML has its own query language: XPath. Since XML is a tree, XPath allows us to traverse that tree and collect elements and attribute values along the way. The result of an XPath query is thus not an XML document, but a set of elements or values.

Given the structure of XML, it does not come as a surprise that XPath has a hierarchical division as well. For instance, the following XPath query selects all `Creator` elements that are children of any `Work` elements in an `Art` document:

```
/Art/Work/Creator
```

So we first select `Art`, the root element (note the leading slash `/`), then all possible `Work` children, and finally all `Creator` elements that are direct descendants thereof. To select `LastName` elements that are children of any `Creator` element, we can use:

```
Creator/LastName
```

Note how the XPath expression does not start with a slash this time, as we do not start from the root but rather from any possible `Creator` element. However,
this will only select children, i.e., direct descendants, of Creator. To select all descendants of a node, we can specify an axis called descendant:

\[
\text{Work/descendant::LastName}
\]

This will find all LastName elements that are somewhere inside a Work, even if nested within other elements.

Finally, this expression selects all year attributes from Work elements:

\[
\text{Work/@year}
\]

Many more constructs are possible. We can filter elements based on attribute values or the elements they contain, much as you would expect from SQL queries. Bear in mind that XPath queries are executed by traversing the whole XML tree, in contrast with relational databases, which use indexes of the data.

### 4.10 Data- versus narrative-centric XML

One of the reasons why there are so many misconceptions about XML is the fact that it can be used for a wide range of purposes. Even if there are a lot of concrete projects and applications which do not fall exclusively in one of the two categories, it is important to distinguish the data- and narrative-centric approach to XML. These two categories largely coincide with how the two different communities we talked about in the beginning of the section understand XML. The IT community has a data-centric view, in the sense that XML is used to define a structure, which is then filled up with data. The example of a Simple Object Access Protocol (SOAP) message below illustrates this approach. The XML file allows to facilitate, in an automated manner, the communication of a specific value between two computers, here the insurance value of Guernica at the Reina Sofia Museum, in a structured format.

```xml
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header/>
  <soap:Body>
    <m:GetInsuranceValue xmlns:m="http://www.example.org/insurance_value">
      <m:Insurance_value>DE00050</m:Insurance_value>
    </m:GetInsuranceValue>
  </soap:Body>
</soap:Envelope>
```
The digital humanities and computational linguists tend to have a narrative-centric vision of XML, in the sense that XML is used to insert some level of structure in documents. The Text Encoding Initiative (TEI) is a classic example of the narrative-centric approach. Below you can find an excerpt from a TEI example file from Wikipedia. The choice tag can be used to represent variants of the same section of text. In the example below, choice is used to indicate an original and a corrected value and to differentiate an original and regularized spelling:

```xml
<p xml:id="p23">
  Lastly, That, upon his solemn oath
  to observe all the above articles,
  the said man-mountain shall have a daily allowance of
  meat and drink sufficient for the support of
  <choice>
    <sic>1724</sic>
    <corr>1728</corr>
  </choice>
  of our subjects,
  with free access to our royal person, and other marks of our
  <choice>
    <orig>favour</orig>
    <reg>favor</reg>
  </choice>.
</p>
```

The fact that XML can accommodate both approaches is due to the fact that XML was conceived to be both human- and machine-readable. This feature of XML is one of both its biggest advantages and its biggest pain points. Any XML file can be opened and modified in a text editor. In theory, you could describe your entire collection in all of its detail by just using Notepad. This platform and application independence makes it particularly easy to exchange XML documents between heterogeneous environments. XML files are in this sense one of the best serialization formats that make the case for non-binary files.

### 4.11 Change

Most changes in XML are difficult and should be carefully considered, as they need to propagate to different documents. In the case of relational databases, changing the structure was cumbersome, but nonetheless always limited to a single system. With XML, if a document format evolves, there are two main options:
• The change is completely backwards-compatible; for example, adding an optional element. Existing documents can then remain as-is, and parsers can be extended. However, this needs careful planning in advance, and is not always possible.
• The change is not backwards-compatible; for example, renaming a tag or changing an attribute into a child element. We distinguish two sub-options:
  — Through schema versioning, different document structures can be supported. Existing documents do not have to change, but parsers must support the different versions (for example, one with the old element name and one with the new name).
  — The change is breaking and existing documents and parsers have to be updated to conform to the new schema. This leads to maximal consistency, but many modifications must be carried out (for example, all documents and parsers have to switch to the new name).

The first option is clearly optimal, but only applies to certain cases. In general, change is difficult and thus progresses slowly. Between changes, data cannot be stored optimally.

4.12 Why do IT people prefer JSON?
According to the followers of the XML hype around the year 2000, XML was soon going to take over the web (and then the world). There are several reasons why this did not happen – even though XML is still very popular in many application domains. First, there is XML’s verbosity. By design, XML tries to be as explicit as possible, but this sacrifices clarity in the end. For many XML documents, it is difficult to understand what is going on with a single look. There is also a lot of repetition in the markup.

More importantly, the web has witnessed an enormous growth of JavaScript applications. At first regarded as inferior to compiled languages such as Java (which has many XML-driven parts itself), it soon became clear that JavaScript’s dynamic nature made it a perfect fit for the web. While JavaScript can parse XML, it also has a native format to express data: the JavaScript Object Notation or JSON, with a more terse syntax that focuses on ‘just getting things done’. As a result, we should not expect much portability between different contexts. However, data exchange over the web between clients and servers, and between different applications, happens mostly in JSON. Severance (2012) neatly describes this oddity of the winning underdog. An example JSON fragment of a work could look like this:

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
Note that JSON has also a hierarchical structure; in fact, the above document translates directly into XML (with the exception that JSON does not offer distinct structures for child element and simple attributes).

4.13 Does XML make your data smart?

In the introduction of this section we mentioned the high expectations collection holders have of XML, due to the common belief that XML makes your data smart. We hope to have demonstrated throughout the section that XML is nothing more, but also nothing less, than a standardized syntax to encode data in a structured manner. The semantics of data can be made explicit through the use of XML elements, but it is important to realize that outside the community which defined the elements, the meaning of the elements is not explicit. The use of namespaces does offer the opportunity to share amongst different communities the same element, but this practice mostly applies to a limited set of the elements. So at the end of the day, even with our metadata in a more portable format we are still confronted with the same problem: if we want other people to re-use our metadata, they are forced to study the schema and documentation describing their semantics.

5 Linked data

We have travelled all the way from previous data models to come to this specific point. Relational databases and XML both offer wonderful possibilities to manage structured metadata, but they also have the big drawback that you need to understand the schema describing the structure and interaction between the data. This is exactly where our last data model comes in.
5.1 The semantic web vision

Before we get into details regarding the data model, let us first fully understand why exactly we need to bypass the problems associated to the re-use of locally defined semantics. The vision behind the semantic web was born out of the frustration of having only human-readable information on the web, which restricts the ways in which software can help us find information. For instance, keyword-based search works well for terms such as ‘Picasso’. Queries such as ‘paintings by Picasso’ are already more difficult, since pages can use different wording. But without an interpretation of a page’s content, queries such as ‘paintings by artists who have met Picasso’ are impossible. In the semantic web vision, the web also becomes accessible for software agents instead of containing only human-readable information (Berners-Lee, Hendler and Lassila, 2001). It enables a vast array of novel applications by making information machine-interpretable.

5.2 RDF

By adopting an extremely simple data model consisting of triples, data represented in Resource Description Framework (RDF) become schema-neutral. An RDF triple consists of a subject, predicate and an object, as seen in Figure 2.1. This allows for maximum flexibility. Any resource in the world (the subject) can have a specific relationship (the predicate) to any resource in the world (the object). There is no limit on what can be connected to what. This model allows us to express statements in a straightforward way, such as for example the statement that Jeff Koons is the artist who created the work ‘Puppy’:

:Jeff_Koons :created :Puppy.

Figure 2.4 represents some of the metadata of the example we have been using throughout this chapter. By simplifying to a maximum the data model, all of the semantics are made explicit by the triple itself. By doing so, there is no longer a need for a schema to interpret the data. Within the world of databases and XML, only the data conforming to the rules defined in the schema may exist and be encoded in the database or XML file. With RDF, you just make statements about facts you know, but these statements might interact with statements made outside your information system. This data model allows heterogeneous data to connect and interact. For instance, in Figure 2.4 you can see two pieces of metadata which were previously not mentioned.

Note, however, that schema-neutral does not mean that no schema-related issues remain. Any piece of data still needs to be expressed in a certain vocabulary, and each vocabulary has its own way of expressing things. The main difference between RDF and XML and other technologies is that in RDF everything is self-describing: each vocabulary is expressed in terms of other
vocabularies. In order to extract meaning from a given RDF fragment, the unique identifiers of each used resource allow its definition to be looked up. This look-up mechanism is enabled by the principles of linked data, which is the topic of the next section.

5.3 The linked data principles

The implementation of the RDF model in the open and distributed context of the web is based upon their capability to issue identifiers for subjects, predicates and objects, which can be freely re-used. Software is then able to interpret this information, because the identifiers create unique meaning, as opposed to the names of columns in databases or elements in XML, which only have local significance and change from application to application.

However, the semantic web was mainly developed from the artificial intelligence (AI) standpoint. Ever since the 1960s, the AI community worked on automated reasoning, expert systems and intelligent agents. Underlying all of these fields and applications is a core belief in the power of logic to formalize all aspects contained within an information system. Chapter 4, in section 4, will come back to the reasons why this vision is currently deemed unworkable on the scale of the web.

In order to move forward with a machine-readable web, Berners-Lee (2006) drastically reduced the ambitions of the full-blown semantic web and came up with the *linked data principles*. These four rules specify a simple way to format data so it can be interpreted by software:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the

![Figure 2.4 Illustration of how to use triples to express metadata](image-url)
standards (RDF, SPARQL).

4 Include links to other URIs so that they can discover more things.

As you can see, these principles require a clear understanding of URLs and URIs. A URL is a *uniform resource locator*, which, as the name says, enables to locate resources in a unique way. The most widely known URLs are those used on the web; they start with http: or https:. Given any such URL, your browser is able to locate the underlying resource, no matter where it is physically stored. A URI, *uniform resource identifier*, is a generalization of the concept that permits resources anywhere in the universe to be given a unique identification. However, not all URIs are URLs; for instance, the URI urn:lex:eu:council:directive:2004-12-07;31 uniquely identifies a European Union directive, but does not directly give its location.

Let’s now look at the role of URIs and URLs in the linked data principles. The first principle demands unique identification for each concept, and URIs are the most appropriate mechanism to provide this. Additionally, the second principle states that these identifiers must be HTTP URIs, in other words URLs on the web. The third principle asks for the representation of the resources identified by those URLs by using standards, such as the machine-readable format RDF. Finally, the fourth principle makes sure that data contains links to other data, allowing software agents to look up related information.

5.4 The central role of URLs

Remember how XML namespaces uniquely identified elements and attributes. With linked data, URLs are used to uniquely identify concepts. For example, a more meaningful way to express the fact that Jeff Koons created ‘Puppy’ is the following triple:

```xml
  <http://purl.org/dc/terms/creator>
  <http://viaf.org/viaf/5035739>.
```

In this example you see that the artist Jeff Koons is identified with the URL of his authority file available on the Virtual International Authority File (VIAF) website. What is the added value of using the URL instead of the string of characters ‘Koons, Jeff, 1955–’? Rules have been developed for decades to formalize the spelling of names in authority records, and in order to disambiguate with other people with exactly the same name, his date of birth has been added. Therefore, one could think that the text string serves well as an identifier. Imagine, however, what needs to happen if Jeff Koons dies in 2025? All of the
metadata which used the text string to designate the name of the creator will need to update the creator field to ‘Koons, Jeff, 1955–2025’. Instead, if the VIAF URL is used, the information only needs to be updated centrally in the VIAF authority file, but the URL as such does not change. From the moment the date of death has been added, this new information will become automatically available to everyone who uses the VIAF URL as an identifier for Jeff Koons. The fact of looking up more information about a subject through its URL is called dereferencing. Chapter 5 will explain the use of URLs for both virtual and real-world resources.

The basic condition for this approach is a stability of the identifier, and URLs tend to have a very bad reputation on that level. The URL used to identify the work ‘Puppy’ is simply the URL of the record displaying the metadata of the object. But what would happen if this work is transferred from the Bilbao to the Venice Guggenheim museum? This would imply that the URL loses its validity. To avoid such trouble, Chapter 6 discusses sustainable URLs.

5.5 Serialization

As the initial semantic web vision was launched in 2001, it comes as no surprise that the first standardized syntax was based on XML, and consequently named RDF/XML. Unfortunately, RDF/XML inherits the verbosity of XML as well, resulting in a serialization format that admittedly can be parsed by an XML parser, but is hard to follow and understand. Therefore, the Turtle syntax was developed, in which triples are native elements. Turtle is currently in the final stage of standardization, and is bound to take the place of RDF/XML.

The triples above were expressed in Turtle, but here we will review its syntax in more detail. In Turtle, triples are serialized by separating each of the components (subject, predicate, object) by white space and ending it with a dot. URLs are surrounded by angle brackets.

Since URLs can be rather long, it includes an abbreviation mechanism through the @prefix directive:

@prefix gh: <http://guggenheim.org/new-york/collections/collection-online/artwork/>.
@prefix dc: <http://purl.org/dc/terms/>.
@prefix viaf: <http://viaf.org/viaf/>.


We first define three prefixes, consisting of certain characters ended by a colon, which can subsequently be re-used. This makes the actual triples shorter and
easier to understand, and can also eliminate possible mistakes in the URL by avoiding duplication.

Multiple statements about the same object can be written tersely by using a semicolon if the subject is repeated, and a comma if the subject and predicate are repeated:

```
gh:48 dc:creator viaf:5035739;
   dc:title "Puppy".
viaf:5035739 :influencedBy viaf:15873,
       viaf:95794725.
```

The above fragment states that the creator of the artwork is Jeff Koons (viaf:5035739) that its title is ‘Puppy’, and that Jeff Koons is influenced by Pablo Picasso (viaf:15873) and Ed Paschke (viaf:95794725). In addition to the use of semicolons and commas, we note two other things. First, the word ‘Puppy’ is surrounded by double quotes, as it is not a URL but a literal value. RDF includes literal values in its model as well, as some properties eventually do not point at another object but rather at a non-decomposable value. Literal values can have an associated type (such as string, number, or date), and in case of a string, a language code (such as en-us). Second, the predicate :influencedBy has an empty namespace prefix, which indicates that it is local to the current document. This is a convenient way of introducing new concepts in a document, which are then defined in terms of other properties later on.

5.6 Search and retrieval

Like relational databases and XML before, RDF also comes with its own query language, SPARQL. Queries in SPARQL are based on graph patterns: the form of the desired data is described in a WHERE clause. A simple SPARQL query is the following one:

```
SELECT ?predicate ?object WHERE {
}
```

This searches for triples that have Picasso as subject and any predicate and any object. The question mark before a word means that it is a variable. Out of those triples, the query will return the predicate and the object.

We can be more specific as well. For instance, if we want to find works by Picasso, we can use the following query. Note the use of prefixes to abbreviate common terms.

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
This illustrates the simplicity of the linked data model, while at the same time showing its immense power and flexibility.

5.7 Change

We hope you understand an essential feature of the RDF data model by now: that all of the semantics of the data are made explicit through the model itself. In a sense, we have come full circle with a return to the idea of a flat file, if we think of a collection of triples contained in a single file, composed of three columns with the headers subject, predicate and object. Confronted with a new reality which needs to be handled, new triples are simply added. However, this comparison does not do justice to the RDF model, as the strength of triples is that every value points to other triples that have this value as subject or object. The context of every row is thereby augmented by other rows. Change in RDF is therefore supported easily: adding new data comes down to adding new triples, without needing to alter the existing data or structure. This gives the data maximum flexibility, at the cost of dealing with an open world. Whereas databases are guaranteed to give you all the data they have, finding all facts about a concept is more complicated with RDF, as different identifiers might be used for the same thing. Nonetheless, when such issues are managed properly, for instance by creating sufficient links between datasets, schema-neutrality can be a very powerful concept.

6 Conclusion

This chapter provided an answer to the question of why we need linked data. The answer might seem self-evident and straightforward: in order to link data across the web. To achieve this goal, we need to be able to interconnect data across independent islands. We use the word ‘islands’, as each information system is modelled for its particular needs and application domain, resulting in systems that cannot hold hands with one another in an automated manner. For sure, it is easy to embed a link in your collection database which points to the record of a similar object from another institution. But this requires you to know how to access the database of the other institution, to know what fields are used to describe the object. Once you have found the record to which you want to link,
you need to embed its URL manually within the record of your database. We cannot reasonably perform these actions manually for all of our collection items. We therefore need to think about how we can automate the linking process.

6.1 Understanding trade-offs

In order to understand the obstacles to the automation of linking, this chapter gave a comprehensive overview of the most recurrent data models used to build the current islands of metadata. Hopefully the red thread between the four data models has appeared clearly. A trade-off has to be made between the complexity of the data model and the ease with which the outside world can re-use and connect to your data. The collection management databases which are currently forming the backbone of our cultural heritage institutions allow complex data to be stored in a way which minimizes redundancy and dependency. You only need to encode once all the information you have in relation to an artist or some very rare and complex technique which requires a lot of documentation to be understood. The day the artist dies, you only need to update the attribute ‘Date of death’ of the entity ‘Artist’, and this update will be shared across all the records pointing to this entity.

However, we have seen that this advantage comes at a cost. A collection management database can easily contain several hundreds of tables, interconnected with relations. Modifications and extra tables are often added in an ad hoc manner in order to fulfil an urgent need, and are often left undocumented. It should therefore come as no surprise that migration operations from one software to the other (or even just to another version of the same software) can be very time-consuming for IT staff, as they need to interpret the database schema to understand how the tables are interconnected.

The development of XML made it easier to share metadata across applications, due to its platform and application independence. In contrast to databases, you do not need any specific software to read and create XML documents. The advantage of being readable both for humans and machines also resulted in XML’s major drawback, namely its verbosity. More importantly, complex data described in XML rely on a schema documenting and prescribing how elements and attributes interact within an XML file. Establishing a consensus inside and outside an institution on how to interpret and update the schema often causes problems.

The last section of this chapter introduced RDF, which we referred to as schema-neutral. The simplicity of the data model (subject-predicate-object) brings back the idea of a flat file, consisting of three columns representing subjects, objects and predicates. No extra documentation or schema is needed to interpret these data, and any new type of information can be added without a need to modify the structure of the data model.
6.2 The law of instruments

‘Give a small boy a hammer, and he will find that everything he encounters needs pounding’. This expression, also referred to as the ‘law of the instrument’, describes how people have a tendency to attribute too much importance to the tool they are using, at the expense of their objectives. One would think that engineers evaluate what data model most suits the needs of an application, and then choose a technology allowing them to implement the model. In practice, the opposite often happens. People build up experience with a specific technology and are not very eager to switch to another tool, as this sometimes requires a substantial effort. Academics and consultants, on the other hand, occasionally tend to get overly enthusiastic about a new technology, regardless of whether the underlying data model is best for the purpose at hand.

6.3 When to use what

This chapter hopefully made it clear that every model has been developed for a specific use. If the only tool you have is a hammer, you treat everything as if it were a nail. This was the case with the use of XML in the context of SOAP, for example. The verbosity of XML is now considered inadequate for data exchange over the web between clients and servers, and its role is taken over by JSON. The current hype on linked data reminds us of the unbounded enthusiasm for XML, in the sense that a lot of applications which are currently built based on linked data technologies could be better and more cheaply realized with a classic relational database. At the end of this chapter, the reader hopefully has acquired a sufficient historical, conceptual and technical understanding of which data model is best for the purpose at hand.

| Table 2.3 Summary of the (dis)advantages of different data models |
|------------------------|-----------------|-------------------------------|
| data model             | (dis-)advantages                                      | use                              |
| tabular data           | + intuitive approach                                  | import and export of data with a simple structure |
| tabular data           | + very portable                                       | import and export of data with a simple structure |
| tabular data           | + technology agnostic                                 | import and export of data with a simple structure |
| tabular data           | - prone to redundancy and leading to inconsistencies  | import and export of data with a simple structure |
| tabular data           | - inefficient search and retrieval                     | import and export of data with a simple structure |
| relational model       | + handling of complex data                            | management of complex data which require normalization |
| relational model       | + optimized queries                                    | management of complex data which require normalization |
| relational model       | + mature software market                               | management of complex data which require normalization |
| meta-markup            | + platform-independent                                | import and export of complex data |
| meta-markup            | + both human- and machine-readable                    | import and export of complex data |
| meta-markup            | - complicated implementation for complex data         | import and export of complex data |
| meta-markup            | - verbosity                                           | import and export of complex data |
| RDF                    | + schema-neutral approach                              | making data available for linking |
| RDF                    | + discovery of new knowledge                           | making data available for linking |
| RDF                    | + loss of normalization                                | making data available for linking |
| RDF                    | + immature software market                             | making data available for linking |

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
model to use in which context.

To conclude this chapter, Table 2.3 summarizes the essential characteristics of every data model we discussed.

The case study at the end of this chapter will now demonstrate how the linked data approach results in a dynamic view of data, allowing heterogeneous information sources to interconnect in a standardized manner. Through the example queries, issues regarding data inconsistency and incompleteness, which is the drawback of this open world approach, will also be underlined. While practising with concrete examples with SPARQL queries, try to reflect specifically on one of the characteristics of the RDF data model that we identified: it is schema-neutral from a conceptual point of view. However, what happens in practice when you want to query a dataset you are not familiar with in SPARQL? We will come back to this issue in the concluding chapter.

7 CASE STUDY: linked data at your fingertips

This case study is slightly different from the others in this book, in that it doesn’t focus on a particular dataset. Instead, we will explore linked data from various sources to get a feeling of the possibilities and limitations in practice. Earlier on in this chapter, we referred to the schema-neutral character of the conceptual RDF data model. The exercises of this case study will help you understand how this theoretical model has been implemented in practice. As you will see, the open-world assumption definitively offers opportunities but can be challenging to implement. To demonstrate this, we will retrieve in this case study metadata on Pablo Picasso in various ways, to understand the capabilities of each data source. First, we will examine DBpedia, which is a version of Wikipedia automatically converted into RDF. Next, we will try queries on Freebase, which is a collaborative linked data source with partly automated input (from Wikipedia and other sources) and partly human input. The difference between DBpedia and Freebase is that Freebase can be edited publicly, whereas DBpedia only has a single automated process. Finally, we will zoom in on Sindice, an index that brings together many large datasets of the linked data cloud and is easily accessible through its front-end Sig.ma.

7.1 DBpedia, the Wikipedia of data

DBpedia is a publicly accessible RDF store with content that is automatically extracted from Wikipedia, the free online encyclopedia. A substantial proportion of articles on Wikipedia have semi-structured data in the form of infoboxes (usually displayed to the right of an article) listing key/value data such as names, birth dates, etc., which are available on DBpedia. Two versions of DBpedia are available: a periodically updated version at http://dbpedia.org/ and a live version
at http://live.dbpedia.org/, which follows the rapid change on Wikipedia. DBpedia currently contains more than 250 million triples. In this case study, we will look at DBpedia and investigate how we can browse and query data.

### 7.1.1 Browsing DBpedia

Finding a topic page is as easy as going to http://dbpedia.org/page/Topic_Name. For instance, the DBpedia page of Pablo Picasso can be found at http://dbpedia.org/page/Pablo_Picasso. At the top of this page, we see its title Pablo Picasso, followed by a short English description. The remainder of the page consists of a long two-column table with properties and values that contain the information we are interested in. Take your time to look around and discover what DBpedia has to say about Picasso. In addition to human-readable abstracts in many different languages, we see many key-value pairs that contain machine-interpretable information. Using the DBpedia ontology, knowledge about Picasso is expressed, such as birth date and place, multi-language labels, influences, spouse, and nationality.

If you wonder where the triples are, well, you can reconstruct them by taking the page’s subject, a predicate from the Property column and an object from the Value column. For instance, one of the triples is:

\[
\text{dbpedia:Pablo_Picasso dbpedia-owl:birthPlace dbpedia:Málaga.}
\]

There are also several properties in the reverse direction as well, indicated by the key ‘is property of’. For instance ‘is dbpedia-owl:parent of’ translates to:

\[
\text{dbpedia:Paloma_Picasso dbpedia-owl:parent dbpedia:Pablo_Picasso.}
\]

As we expect from linked data, we can click through on any value to learn more. If we click on dbpedia:Málaga, we see a page with detailed information on the city. As we explained earlier, even the properties can be examined. Clicking dbpedia-owl:birthPlace reveals that this is a relation between a person and a place. Interestingly, some links go outside DBpedia, thus connecting this dataset to others, something that is not possible with relational databases. For instance, http://data.nytimes.com/N855344257183137093 is indicated as the same resource, with the owl:sameAs relation, and this link leads to Picasso’s data sheet on the New York Times website. This reveals the true potential of linked data.

At the bottom of the page, there are links to view the data in different formats. The ‘N3/Turtle’ link leads to an RDF serialization that can be interpreted by software. You might notice a lot of strange-looking sequences in the file which take up a large amount of space. They are escape sequences for non-ASCII characters, such as å, ä, or ö for example, from the full-text abstraction fields.
However, towards the bottom of the file, you will notice more familiar triples such as:

dbpedia:Noel_Rockmore dbpedia-owl:influencedBy
  dbpedia:Pablo_Picasso.
dbpedia:Ben_Shahn dbpedia-owl:influencedBy dbpedia:Pablo_Picasso.
dbpedia:Piet_Mondrian dbpedia-owl:influencedBy
  dbpedia:Pablo_Picasso.

If possible, we recommend that you switch off line wrapping in your editor, so the long lines with escape sequences will simply flow off the screen.

7.1.2 Querying DBpedia

Once we get to know some basic properties of the data, we have sufficient information to start querying it in a more complex way. We have seen some predicates and some objects, which can help us construct queries. We advise you to keep the Picasso page open in one tab while you bring up the SPARQL query interface at http://dbpedia.org/sparql. You are greeted by the following default query:

```
SELECT DISTINCT ?Concept WHERE { [] a ?Concept. } LIMIT 100
```

We are already familiar with the WHERE and SELECT clauses, and as their names suggest, DISTINCT asks for unique items and LIMIT 100 for only the first 100 results. The [] syntax is a way to say 'any node', like a variable without a name. If we execute this query, we will receive a (quite random) list of 100 Concepts in DBpedia. These are not only DBpedia topics, but also concepts such as owl:Thing and http://schema.org/CreativeWork.

Let's now try a query of our own, to verify if we can get the same information on Picasso as we did when browsing DBpedia. To see all triples on Picasso, enter the following SPARQL query:

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
```

This will show us all triples we saw earlier on the Picasso page. Well, at least those triples that have Picasso in the subject. To find all triples where Picasso is the object, issue the query:

```
```

This yields results like the following:

<table>
<thead>
<tr>
<th>s</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpedia:The_Three_Dancers</td>
<td>dbpedia-owl:artist</td>
</tr>
<tr>
<td>dbpedia:The_Accordionist</td>
<td>dbpedia-owl:artist</td>
</tr>
<tr>
<td>dbpedia:Desire_Caught_by_the_Tail</td>
<td>dbpedia-owl:author</td>
</tr>
<tr>
<td>dbpedia:Olga_Khokhlova</td>
<td>dbpedia-owl:spouse</td>
</tr>
<tr>
<td>dbpedia:Stanley_William_Hayter</td>
<td>dbpedia-owl:influenced</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

The values in the s column correspond to bindings of the ?s variable, the p column to bindings of the ?p variable. To understand where the found information comes from, we substitute s and p in the original WHERE clause. The first row thus belongs to a match of:


To receive just 30 triples, add the LIMIT clause:


If you want to see all predicates used with Picasso as the subject:

SELECT ?p WHERE { dbpedia:Pablo_Picasso ?p ?o. }

Note the omission of the ?o variable in the SELECT clause, as we only want to see the predicates. This yields the following list:

<table>
<thead>
<tr>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a></td>
</tr>
<tr>
<td><a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a></td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td><a href="http://www.w3.org/2002/07/owl#sameAs">http://www.w3.org/2002/07/owl#sameAs</a></td>
</tr>
<tr>
<td><a href="http://www.w3.org/2002/07/owl#sameAs">http://www.w3.org/2002/07/owl#sameAs</a></td>
</tr>
<tr>
<td><a href="http://www.w3.org/2002/07/owl#sameAs">http://www.w3.org/2002/07/owl#sameAs</a></td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td><a href="http://purl.org/dc/terms/subject">http://purl.org/dc/terms/subject</a></td>
</tr>
<tr>
<td><a href="http://purl.org/dc/terms/subject">http://purl.org/dc/terms/subject</a></td>
</tr>
<tr>
<td><a href="http://purl.org/dc/terms/subject">http://purl.org/dc/terms/subject</a></td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
You might be surprised to see that there are duplicates in the result list. How come we have duplicates here when we did not have them in the previous query? The answer is that triples are unique in the triple store, i.e., a triple can only occur once. However, many triples can use the same predicates, and indeed, several Picasso triples use the `rdf:type` and `owl:sameAs` predicates.

To have the unique predicates, we need to add the `DISTINCT` modifier:

```sparql
SELECT DISTINCT ?p WHERE { dbpedia:Pablo_Picasso ?p ?o. }
```

So far, we have received tables of variable values as a result, but what if we want triples? Besides `SELECT`, SPARQL also has a `CONSTRUCT` clause that creates triples instead of variable bindings. For example, this shows all Picasso triples:

```sparql
CONSTRUCT { dbpedia:Pablo_Picasso ?p ?o. }
WHERE { dbpedia:Pablo_Picasso ?p ?o. }
```

These include the following:

- `dbpedia:Pablo_Picasso rdf:type foaf:Person,`  
  `yago:SpanishPotters,`  
  `yago:PeopleFromParis,`  
  `yago:BalletDesigners.`
- `dbpedia:Pablo_Picasso dcterms:subject category:Cubism`  
  `category:Spanish_expatriates_in_France,`  
  `category:Spanish_sculptors,`  
  `category:Modern_painters.`

It might seem strange to duplicate the graph from the `WHERE` clause into the `CONSTRUCT` clause, but they actually signify two different things. The `WHERE` clause tells the SPARQL engine to look for all triples that have Picasso as the subject and to store their predicates and objects in the variables `?p` and `?o` respectively. The `CONSTRUCT` clause instructs the engine to collect all `?p` and `?o` values – regardless of how they were retrieved – and to create triples from them using the specified pattern.

This means that we can choose to generate a different pattern. For instance, suppose that we just want to express that Picasso is somehow connected to the object of the triple, instead of exactly specifying this relationship. Then we can simply do:

```sparql
CONSTRUCT { dbpedia:Pablo_Picasso <isConnectedTo> ?o. }
WHERE { dbpedia:Pablo_Picasso ?p ?o. }
```
This will yield triples such as:

dbpedia:Pablo_Picasso <isConnectedTo> category:Modern_painters.

This allows you to convert the queried data into the form that you prefer. Finally, to receive 100 random triples from DBpedia, try the following:

CONSTRUCT { ?s ?p ?o. } WHERE { ?s ?p ?o. } LIMIT 100

7.2 More complex SPARQL queries

WHERE patterns can be as complex as you like. The most simple case is a single triple. For instance, the birth place of Picasso can be retrieved by:

SELECT ?place WHERE {
}

This turns out to be http://dbpedia.org/resource/M%C3%A1laga. Note the use of escape sequences in the URL to encode the accented character in ‘Málaga’. We can now find all people born in Málaga:

SELECT ?person WHERE {
}

Unsurprisingly, this includes Picasso himself:

```
person ...
dbpedia:Pepe_Romero
dbpedia:Pablo_Picasso
dbpedia:Edu_Ramos
dbpedia:Carlos_Aranda ...
```

We could simplify the same question by describing the pattern in one query with a WHERE clause consisting of two triples:

SELECT ?person WHERE {
}
This will select all people whose birthplace is the same as Picasso’s, without having to specify that exact place. We just instruct the SPARQL engine to find the birthplace for Picasso and look for people with this birthplace at the same time. Were you surprised to see Picasso in the result list? It might seem strange, but this is actually logical: Picasso has the same birthplace as Picasso, hence he is included in the list. Always remember that computers execute what you ask for, not what you intended to ask: Picasso satisfies the query pattern, so his name is returned, even though you already knew this.

We could place further restrictions on this list. For instance, we see many different kinds of people. If we are only interested in artists born in Málaga, we can say:

```
SELECT ?person WHERE {
    ?person a dbpedia-owl:Artist.
}
```

The list has fewer members, and Picasso is still in there (since he’s an artist):

<table>
<thead>
<tr>
<th>person</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
<tr>
<td>dbpedia:Javier_Conde</td>
</tr>
<tr>
<td>dbpedia:Pepe_Romero</td>
</tr>
<tr>
<td>dbpedia:Pablo_Picasso</td>
</tr>
<tr>
<td>dbpedia:Juan_Antonio_Arguelles_Rius</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Let’s ask for people who were influenced by Picasso:

```
SELECT ?artist WHERE {
    ?artist a dbpedia-owl:Artist.
}
```

And let’s see where those people were born, to have an idea of how Picasso’s legacy spread geographically:

```
SELECT ?artist, ?place WHERE {
    ?artist a dbpedia-owl:Artist.
}
```

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
This reveals the following data:

<table>
<thead>
<tr>
<th>artist</th>
<th>place</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpedia:Sarah_Stein</td>
<td>dbpedia:San_Francisco</td>
</tr>
<tr>
<td>dbpedia:Helmut_Kolle</td>
<td>dbpedia:Charlottenburg</td>
</tr>
<tr>
<td>dbpedia:Karel_Appel</td>
<td>dbpedia:Amsterdam</td>
</tr>
<tr>
<td>dbpedia:Karel_Appel</td>
<td>dbpedia:Netherlands</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Note how some artists occur twice in the list, but with a different place. For instance, Karel Appel has a birthPlace of Amsterdam but also The Netherlands, both of which are correct. It might be confusing that they appear twice, but this is because the values are coming from different triples and both answers conform to the query pattern we gave. When we ask for the data as triples, this connection becomes more obvious.

CONSTRUCT { ?artist dbpedia-owl:birthPlace ?place. } 
WHERE {
  ?artist a dbpedia-owl:Artist.
}

Triples indeed better illustrate the connection between the pieces of data:


Another perspective is time: when were the people who were influenced by Picasso born?

SELECT ?artist, ?date WHERE {
  ?artist a dbpedia-owl:Artist.
  ?artist dbpedia-owl:birthDate ?date.
}

This gives the following people:

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
We have a considerably shorter list than when we asked for all people influenced by Picasso. This appears strange, because the amount of people should be the same, no matter when they were born. The issue here is that DBpedia does not contain birth data information for all people. As a result, only people with a birth date are included. In contrast, a relational database would include all people but leave unknown birth dates empty. We can obtain the same behaviour from DBpedia by marking the triple with the birth date OPTIONAL:

```
SELECT ?artist, ?date WHERE {
  ?artist a dbpedia-owl:Artist.
  OPTIONAL {
    ?artist dbpedia-owl:birthDate ?date.
  }
}
```

This will give us all influenced people, some of which have an unknown birth date:

<table>
<thead>
<tr>
<th>artist</th>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpedia:Wifredo_Lam</td>
<td>1902-12-08</td>
</tr>
<tr>
<td>dbpedia:Karel_Appel</td>
<td>1921-04-25</td>
</tr>
<tr>
<td>dbpedia:Piet_Mondrian</td>
<td>1872-03-07</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

To obtain a chronological overview, we can ask DBpedia to ORDER the results:

```
SELECT ?artist, ?date WHERE {
  ?artist a dbpedia-owl:Artist.
  OPTIONAL {
    ?artist dbpedia-owl:birthDate ?date.
  }
}
ORDER BY ?date
```
This will list artists without a known birth date first, followed by an ordered list of those with a known birth date. Finally, we might be interested to dig even deeper. All artists in the current list were influenced by Picasso, but who did those artists influence?

```
SELECT ?artist, ?influencedArtist WHERE {
  ?influencedArtist dbpedia-owl:influencedBy ?artist.
}
```

The SPARQL engine first looks for all ?artist matches who were influenced by Picasso. For each ?artist value, it looks for ?influencedArtist matches who were influenced by ?artist:

<table>
<thead>
<tr>
<th>artist</th>
<th>influencedArtist</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpedia:Karel_Appel</td>
<td>dbpedia:Jan-Hein_Arens</td>
</tr>
<tr>
<td>dbpedia:Piet_Mondrian</td>
<td>dbpedia:Charmon_Von_Wiegand</td>
</tr>
<tr>
<td>dbpedia:Piet_Mondrian</td>
<td>dbpedia:Robert_Cottingham</td>
</tr>
<tr>
<td>dbpedia:Georges_Braque</td>
<td>dbpedia:Piet_Mondrian</td>
</tr>
<tr>
<td>dbpedia:Georges_Braque</td>
<td>dbpedia:Byron_Galvez</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Now is the time to reflect on what we have achieved. The above query selects people influenced by people who were influenced by Picasso. We sincerely challenge you to try finding this information on Google (do let us know how many searches and clicks you needed).

One important remark though: the returned information by DBpedia is *incomplete* and likely *incorrect*. There are two reasons for incompleteness. First, not all data might be there. Some artists might not be in DBpedia, others might not have their influences listed. If DBpedia does not know an artist’s influences, that result will simply not be included. Second, SPARQL endpoints are not obliged to return all results. RDF has an open-world assumption: just as on the web, we are never sure that all information is there. Hence, the SPARQL engine gives its best effort to find your query results, but without the guarantee that they will be complete. You can try to formulate a query that gives you all the data from DBpedia (it’s really easy in fact), but you will notice that you only get a few thousand results. It is not that the other triples are not there; it is just that the SPARQL engine decided it has worked hard enough already. The reason for incorrectness, apart from missing information, is that the information in DBpedia stems from the online encyclopedia Wikipedia, which is edited by volunteers. That information might contain errors, or might be outdated. The SPARQL endpoint of
DBpedia Live (http://live.dbpedia.org/sparql) might give you more up-to-date results, but unfortunately, no correctness guarantee either. However, results from a search engine such as Google can also be incomplete and incorrect.

7.3 Freebase, the community-curated database

7.3.1 Browsing Freebase

While DBpedia data stems from Wikipedia, an encyclopedia that can be edited by anyone, DBpedia data itself is not publicly editable. Freebase (http://freebase.com/) is a database of linked data where users can add and edit information directly. That does not mean that Freebase contains only human-added data: many entries are loaded automatically from external sources (such as Wikipedia and Netflix) by specialized software tools. However, human curation is an important aspect of the Freebase philosophy.

Freebase is not based on RDF, but its contents are still considered linked data because a triple-like model is followed. Data is organized by topic, and typed relations to other topics can be added; topics and triples are identified by URLs. In contrast to DBpedia, each piece of data also contains fields detailing by whom and when it was created. Furthermore, facts are grouped by types. For instance, people have the type People, which contains properties such as Date of birth, Gender, and Children. These properties are not always filled in for all members of the People type; they just provide an easy template for editors to encourage them to use the correct properties.

Let's have a look at the page for Pablo Picasso on Freebase. You can visit it directly at https://www.freebase.com/m/060_7 or search through the homepage https://www.freebase.com/ for the artist's name. This page starts with an image and an abstract of the topic, as well as various links to pages on the web about the same topic. In the ‘Properties’ tab below, we see all data on Picasso, grouped by type that belong to categories. The first type is Topic and lists general information such as name, description, website and notability. These properties apply to all topics on Freebase. Scrolling down the page, we note other types such as Person and Visual Artist, as well as Literature Subject and Film Actor. It might seem strange to label Picasso as a film actor, but this is actually merely a grouping of properties per topic. As we expect from linked data, clicking values leads to the information pages about the subjects. Hovering over types and relationships also reveals their properties.

On the top of the page, we can see different tabs. The default view is the Properties view. The I18n tab (an abbreviation of ‘Internationalization’) shows attributes with multi-language fields, such as transliterations of Picasso’s name in different languages. The Keys tab lists different identifiers of the subject, together with their namespace and who created them. Finally, the Links tab shows a tabular overview of all properties of the subject, together with the
person who added or last edited them and the modification date. This can also
serve as a history of all data about the topic.

Freebase is more human-targeted than DBpedia, as the pages are designed to
be browsed and edited easily, whereas DBpedia offers primarily data tables. This
is already apparent from Freebase's homepage, which allows you to navigate data
in an interest-driven way. Similarly to DBpedia, data about topics is also offered
in RDF, although this might not be obvious at first sight. Starting from the regular
URL for a topic such as https://www.freebase.com/m/060_7, you can obtain the
RDF version by changing the URL to http://rdf.freebase.com/rdf/m.060_7.

7.3.2 Querying Freebase

Freebase does not support SPARQL queries; rather, it uses its own query
language, MQL, which is based on JSON. The reason for this is historical;
SPARQL was not yet standardized when Freebase came into existence. The
query form of Freebase is located at https://www.freebase.com/query.

The following query retrieves all personal data about Pablo Picasso:

```json
{
    "name": "Pablo Picasso",
    "type": "/people/person",
    "*": null
}
```

The idea is to pass a template to the query engine, and all empty fields will be
filled out in reply. The fields name and type narrow the query down to a single
topic. The field * ('all') with value null ('a single unknown value') instructs
Freebase to select all values that belong to this topic. The result is a JSON
document similar to the following:

```json
{
    "result": {
        "place_of_birth": "Málaga",
        "id": "/en/pablo_picasso",
        "parents": [
            "José Ruiz y Blasco",
            "María Picasso y López"
        ],
        "gender": "Male",
        ...
    }
}
```

The results you receive depend on the type you specify. If we want to retrieve
properties about Picasso’s work as an artist, we set the type accordingly:

```json
{
  "name": "Pablo Picasso",
  "type": "/visual_art/visual_artist",
  "*": null
}
```

This then results in the following data:

```json
{
  "result": {
    "type": "/visual_art/visual_artist",
    "id": "/en/pablo_picasso",
    "name": "Pablo Picasso",
    "art_forms": [
      "Painting",
      "Sculpture",
      "Ceramic art",
    ]
  }
}
```

If we are interested in specific properties, we can specify them in the template. For instance, to only show the artworks by Picasso:

```json
{
  "id": "/en/pablo_picasso",
  "type": "/visual_art/visual_artist",
  "artworks": []
}
```

The two square brackets `[]` denote an empty list (‘multiple unknown values’), which will be filled out by the query engine:

```json
{
  "result": {
    "id": "/en/pablo_picasso",
    "type": "/visual_art/visual_artist",
    "artworks": [
      "Guernica",
      "Garçon à la pipe",
      "Les Demoiselles d’Avignon",
    ]
  }
}
```
Note that a null value would not have worked here, because there is more than one artwork.

If we want more details on the artworks, we have to extend the template. For instance, to retrieve all artwork properties:

```json
{
    "id": "/en/pablo_picasso",
    "type": "/visual_art/visual_artist",
    "artworks": [{ "*": null }]
}
```

Inside the list, we place a single template object whose properties will be filled in. This returns an extensive list of all works and their properties. If we are only interested in specific properties, we can name them as we did before:

```json
{
    "id": "/en/pablo_picasso",
    "type": "/visual_art/visual_artist",
    "artworks": [{ "name": null, "date_completed": null }]
}
```

The templating mechanism thus works on every label of the data. The following query lists all of Picasso's artworks with title and date:

```json
{
    "result": {
        "artworks": [
        {
            "name": "Guernica",
            "date_completed": "1937-06"
        },
        {
            "name": "Garçon à la pipe",
            "date_completed": "1905"
        },
        {
            "name": "Les Demoiselles d’Avignon",
            "date_completed": "1907"
        },
        ...
    }
}
```

The main difference between MQL and SPARQL is that SPARQL still employs...
the triple model (with variables for unknowns), whereas MQL has proprietary templating mechanisms.

7.4 Sindice, the semantic web index

7.4.1 Querying the entire web

So far, we have only been querying single datasets. However, the central idea of linked data is of course to be able to cross dataset boundaries. On the web, we mostly discover information through search engines, so analogously, a search engine for linked data could help us find datasets and navigate to other datasets from there.

Sindice (http://sindice.com) is an index of the semantic web. It collects linked data in RDF and other formats, and helps you discover more resources. For instance, we can inspect the data Sindice has about Picasso by putting Picasso's URL, http://dbpedia.org/resource/Pablo_Picasso, into the text box and clicking ‘Search’. Even though Sindice finds hundreds of matching documents, you might initially be disappointed by the results. While the ranking of results on traditional search engines has considerably improved over the past decades, ranking of semantic data is still in its infancy. However, going through the result pages, we discover various datasets that indeed mention Pablo Picasso.

More targeted searches are possible through Sindice’s SPARQL endpoint at http://sparql.sindice.com/. For instance, we can see what data Sindice has available on Picasso:

```
SELECT ?s, ?p WHERE { ?s ?p dbpedia:Pablo_Picasso. }
```

In addition to a lot of data from DBpedia, we also see triples from other datasets such as the New York Times. The Sindice SPARQL endpoint is also still under development, so not all triples that are available through the search function can be found in the SPARQL endpoint. Another reason we do not see more data is the issue of identity. Indeed, triples in DBpedia are mostly of this form:

```
dbpedia:Pablo_Picasso dbpedia-owl:birthPlace dbpedia:Málaga.
```

However, in the New York Times dataset, triples use another URL to represent Picasso:

```
nytd:N855344257183137093 skos:prefLabel "Picasso, Pablo".
```

This is a common practice in linked data, since there is no central identity
authority. Fortunately, both identifiers are connected together by an owl:sameAs predicate:

nytd:N855344257183137093 owl:sameAs dbpedia:Pablo_Picasso.

Therefore, we can find more triples about Picasso by adapting our query. Instead of demanding that the triple contain the DBpedia identifier dbpedia:Pablo_Picasso, we say that they can use any identifier, as long as it corresponds to the same concept as dbpedia:Pablo_Picasso. In SPARQL, we can express this as:

```sparql
SELECT ?picasso, ?p, ?o
WHERE {
  ?picasso owl:sameAs dbpedia:Pablo_Picasso.
}
```

This instructs Sindice to find all ?picasso identifiers that have a owl:sameAs relation to DBpedia’s Picasso entry, and for each of them, find all matching triples. We now retrieve results from the New York Times, Yago, and several other datasets. This provides an interesting opportunity to harmonize the triples using one single identifier. Therefore, we can construct the triples while explicitly using the DBpedia identifiers as the subject:

```sparql
CONSTRUCT { dbpedia:Pablo_Picasso ?p ?o. }
WHERE {
  ?picasso owl:sameAs dbpedia:Pablo_Picasso.
}
```

Note the explicit mention of dbpedia:Pablo_Picasso in the CONSTRUCT clause, which ensures that all found triples use this identifier, no matter what dataset they originate from.

### 7.4.2 Browsing Sindice through Sig.ma

While Sindice gives you a more raw view on the web’s data, Sig.ma (http://sig.ma/) is an interface built on top of Sindice that lets you collect information about a topic easily. Sig.ma looks through different data sources using Sindice, but groups the information visually together, turning it into a mash-up.
Figure 2.5 shows the result of searching for http://dbpedia.org/resource/Pablo_Picasso on Sig.ma. On the left are the different pieces of information, grouped together by category, starting with images at the top. On the right, we see the sources that contributed to the information on the page. They consist of the original URL we entered, but also all URLs that are mentioned on that page. Scrolling down, we see many information sources, and if we hover above them, the corresponding data sources are highlighted on the right. Similarly, we can hover in the sources sidebar on the left to see which pieces of information they contributed.

However, as you would expect, this mostly shows information associated with the single identifier http://dbpedia.org/resource/Pablo_Picasso and so a lot of potential sources are not included. Therefore, we can also search for the artist by typing ‘Pablo Picasso’ in the search box, which will lead to much more data. However, Sig.ma text search is a little liberal, so some data might only be marginally relevant to Picasso. Even worse, some data is simply incorrect because the topic was not matched precisely. Here are some examples we found:

- Picasso is created by ‘darj33ling’. Upon closer inspection of the data source containing this fact, it turns out that this actually refers to a slide deck titled ‘Pablo Picasso’, created by a user account ‘darj33ling’.
- Picasso is 425 pixels wide, because a photograph of him is.
- Picasso is a product or service. This seems to come from an online shop that sells puzzles with Picasso’s artworks on them.

For more Facet books and to order online visit: http://www.facetpublishing.co.uk
This indicates the kind of problem that can occur when automatically combining data from different sources. It also shows the importance of having URLs as identifiers on the semantic web instead of plain text. Fortunately, Sig.ma allows you to discard sources you do not trust by removing them from the right sidebar. Additionally, you can add new data sources there. Sig.ma will extract data from them on the fly and categorize it on the left.

All in all, Sig.ma is an exciting visualization of the possibilities of linked data, because it connects so many sources together. If we understand its limitations and are careful with selecting sources, very meaningful results can be generated. Do try to put your own name in the search box, as there might even be linked data sources about you that you weren’t aware of. And if you have your own data, you can submit it to Sindice so it will be included in Sig.ma results.

Notes
1 We based our overview of data modelling on chapters from Ramsay (2004) and Segaran, Evans and Taylor (2009).
3 You are not strictly required to add the PREFIX declarations here, as DBpedia inserts them automatically for its common prefixes. However, not all SPARQL endpoints support this and so, in general, always include all declarations (even though we will not repeat them in this book due to space constraints).
4 At least, a triple can only occur once in the same graph. Triple stores might contain different graphs, but this is a different story altogether.

References
Instructions

Digital Yoknapatawpha Project: Entering Data (January 2015 Revision)

Here I’ll try to explain both the larger concepts that govern the way we’re turning Faulkner’s texts into “data,” and the specific steps by which you’ll use the Drupal program to enter each piece of data into the project’s database. I hope I don’t sound too bossy, but the success of the whole project depends on the database, and a good database requires accuracy and consistency, so it’s crucial that we all use the same guidelines.

We are, however, still at an early point in the process of building DY, and many unanswered questions remain about what we should be doing, including how our databases should be defined. Event Keywords, for example, is a category we are still in the process of defining together. There are, for example, two major categories where we will work together to decide what we should do. I’ll explain this further below, in the sections on “Notes” for the Event category.

And more generally, you should always feel free to raise questions about all the categories and fields, as we did at our meeting in Charlottesville in July. (This revised set of Instructions includes the additions and refinements we agreed on at that meeting.) There will be multiple ways to share your concerns and suggestions with the group over the next several years, and there is no limit on the number or kind of questions we can try to answer together. So even as you follow these instructions and enter data according to the current Location, Character and Event fields, please keep track of whatever you think needs changing, adding, improving. Both our database and these instructions are very much works in progress.

I’m sorry these instructions are so long, but we learned from the first set of stories that there’s a lot that needs to be spelled out. Please read through the instructions once, in full, to get larger picture, including a sense of the conceptual issues we still need to resolve. But you don’t have to memorize anything. Once you start entering data you’ll be able to use the HELP buttons on the Drupal entry forms for each of the specific Locations, Characters and Events fields.

And if there’s anything in these Instructions that is unclear, or incomplete, or unnecessarily annoying, let me know that too, please. My prose is very revisable! Thanks. Steve (October 2012)

GENERAL NOTES:

Each team should come up with its own plan for collaborating, but it is important that every piece of data be seen by the whole team, and agreed on by at least 2 members. Who are the Characters in your story? and what is each one’s Rank? etc. – all these decisions should be ratified by the team.

If a team simply cannot agree on what data to enter – say, what Class a character belongs to – the advisory editors and I will be there to resolve the dispute (I hope!).

Based on her experience editing “Spotted Horses” this past spring, Dottie Dye suggested we come up with a template for each of the 3 categories, so we’ve done that. Using the templates isn’t a required step. Teams can confer about the data any way you want before using Drupal. The process in Drupal, however, is set up so that someone entering data cannot SAVE a new entry until all the required fields have been filled in. Working the fields out ahead of time with templates should make the process go more smoothly.

An alternative procedure could be to have one editor enter all his/her data for Locations into Drupal, where the other editors could view and, whenever there is a question or disagreement, comment on the choices.

View and print each template by following “Download data entry template” link at the top of each main category.

You have to log in to Drupal to create or revise data. This allows the system to track authorship of each piece of data.
In these instructions I discuss the three main categories of data in the order in which it makes the most sense to enter them: 1. Locations, 2. Characters, 3. Events (you’ll see why you should work in that order as you keep reading). For each piece of data in your story – Location, Character, or Event – you’ll start by clicking on the “Add a new [Location|Character|Event]” link under the “Download template” link.

When entering data, it’s a good idea to have two or three browser windows open to the Drupal site at the same time. That way you can keep the page on which you’re entering data open on one window, while using the other(s) as necessary for cross checking, getting x- and y-coordinates, and so on. Drupal isn’t the easiest program to move around in, so it’s good to keep the page you’re filling in open, and use other windows to move around. (Ex: I realized entering dates for Flags Events that I often lost track of the date of preceding event; being able to check that in a different window was much easier than saving the current event, clicking back to the previous one, etc.)

And once you’ve “Added” all your entries in each category, and are revising or reviewing, here are a couple other program functions that will probably come in handy.

The front pages for each main category (Locations, Characters, Events) display all the items in the database for that category, by Text and Name, but also along with a few other kinds of data (exs: Pg, i.e. Page Event Begins, with Events; or Date_of_Birth with Characters, etc). Currently each category sorts by one of these fields (exs: Text in the case of Locations, or Name in the case of Characters). You can ask Drupal to re-organize the lists by other fields. For example, when you’re editing the Characters you created for a story, by clicking on the word Text in the first column, you can gather all the characters in your story together. Clicking on any field in the top row will organize the list by that field.

And an important extra feature is the text filters at the top of each category (Locations, Characters, Events), which again will be useful as you revise or review the entries you created. By using the filter’s pull down menu and clicking on your text, you’ll eliminate all the data from other stories. This is especially useful when reviewing and revising Events. After you use the filter to segregate only the Events from your text, you can click on “Pg,” the second column, and sort all your story’s Events into narrative sequence, from beginning to end of story.

This sounds more complicated than it is, and you can practice with the data that’s already in Drupal – i.e. try using filter, sorting by the various fields, etc. Just make sure you don’t edit data that any other team has created!

We have agreed to use Faulkner’s Collected Stories and Uncollected Stories (ed. Blotner) as source texts for the short stories, and the Viking International paperback texts (“Corrected by Noel Polk”) as the source texts for the novels. If your team feels there’s another version of a story that should be used instead, please contact me so we can discuss your reasons. As a last step in editing your story, you’ll write an “About the Text” note, and there you can discuss what other versions of a story exist, what differences might be, etc.

Brief quotations from your story can enrich a Location or Character description, or a summary of an Event, so feel free to use them. Quotations should all be identified by page numbers in parentheses, right after the quotation.

When you’ve finished each item, hit the SAVE button at the bottom of the form. The Drupal system will tell you if there are any problems. If not, you then go back to the “Add a new...” link to begin entering the next Location, Character or Event.

---

Instructions for Locations Data Entry

What is a “Location”?

In the Summer 2012 the editors of the first 4 stories and I came up with a consensus answer to this question which was re-confirmed at our Summer 2013 meeting. This is one of those things that we can discuss changing, but in the meantime, please use this definition:

(1) Places in or outside Yoknapatawpha where Events in your story occur are all Locations.

(2) In addition, all places in Yoknapatawpha county that are clearly identified in your story are Locations. (Ex: in “Spotted Horses” the narrator talks about trying to sell a sewing machine to Mrs. Bundren; narrative never takes readers there, but “Bundren Farm” is a Location that goes on the map.)

(3) In addition, places outside Yoknapatawpha that play a significant role in your story are Locations. (Ex: the spotted horses and Buck come from Texas, Flem and Eula spend a year in Texas, so even though narrative never takes reader to Texas, Texas goes on the map.)

(4) But places outside Yoknapatawpha that are simply mentioned in passing are not Locations. (Ex: Quentin remembers that his parents brought Jason a souvenir back from the Fair in St. Louis. St. Louis does not go on the map.)

---

http://faulkner.drupal.shanti.virginia.edu/content/instructions
This still leaves a lot of room for interpretation: for example, “Kentucky” in The Sound and the Fury? Gerald Bland’s life there is often talked about, though narrative never goes there, not even through Quentin’s memory. I would say “Kentucky doesn’t go on the map,” but teams should resolve this kind of question together for their particular text, and later we’ll talk about how make definition as consistent as possible.

A Note on Maps – We create a new map for each story or novel, so please be thinking about how the map for your text should look – what roads to add or subtract, say, or which landscape features need shifting, etc.

Creating Locations:

The first step is to agree with your co-editors on the list of Locations, according to the definition above. Then check under the Location Keys tab to see which of those locations have already been named elsewhere in the project.

The Locations Key is called a "lookup table.” You will create a Location entry for every location in your story. But the purpose of the Location Keys is to make sure places have consistent names, so that they can be combined and searched across texts. "Sartoris Plantation," for example, requires a name and code number that will be constant through all the data tables. If one team calls the "Sartoris Plantation” the “Sartoris Place,” a search for the first won’t turn up the second; that's how literal computers are. (We’ll use this same process for Characters, as you’ll see when you get to that category.) So: the Key provides a number and name for, say, the Sartoris Plantation in all the fictions.

“Sartoris Plantation” is already on the list, so you won’t add it to the Key. For all your locations that aren’t in the Key, however, use the “Add a new location key” link to create a Key entry. Once you’ve done this, you can begin adding the Locations themselves, according to the specific instructions below.

If your team came up with, say, 8 locations for your text, you will “Add a new location” 8 times. (This includes, for ex, the Sartoris Plantation, if it’s in your story – Key provides the name, but you still have to create the specific Location entry for “Sartoris Plantation in your text.”) You don’t enter a Location more than once, though; i.e. in Flags in the Dust the narrative keeps returning to the Sartorises (and each of those returns will be a separate Event, on that database), but there’s only one “Sartoris Plantation in Flags in the Dust” under Locations.

*Required fields are indicated with asterisks. This means you must make an entry in that field.

*Text
Use pull down menu. Choose the story you’re editing.

*Location Key
Choose from the lookup list. If the location you need isn’t there, on a second browser window open to Drupal go to Home, and select Location Keys from tabs across top.

Look again at list of locations on the Key, to make sure the one you want isn’t already there. If it isn't, make a note of the “Code” number of the last location on the list.

Then click on “Add a new location” from the top part of this page. All you enter here are two things: (1) a Code, which is the next number after the last one already on the list, and (2) a Name. The Name should be short but descriptive.

For unique places (i.e. the Compson Place) don't include the title of your text in the Name; the Name you give the location here is the one it will have whenever the same location appears in other texts too.

For non-unique or generic kinds of places (i.e. a Campsite or Bridge, etc.) do include name of story (ex: "Campsite in Barn Burning").

*Display Label
This is what users will see when they hover their cursor over a Location. Labels should be short and direct. They should not include name of text (which users will already know).

*Type
Choose from drop down menu.

“Event” is what you use for an episode that isn’t attached to a recognizable structure – Bayard’s car accidents in Flags in the Dust, for example. Use "Other Structure" for physical objects that aren't covered by any other Type on the list (a water tower, for ex, or a farm outbuilding like a cotton house, etc.); this will put a generic "Other Structure" icon on the map, so label should include a specific identifier ("De Spain's Barn," for ex).
For Locations that can’t appear on a map of Yoknapatawpha – “Memphis,” “Paris,” etc. – use “Out of Yoknapatawpha” as the type. These Locations create such a large set of separate issues that here I’ll just say, if your text contains such Locations, add them according to the following directions, and your team and I will then talk about how to put them on our map. For Out of Yoknapatawpha locations, you will not provide x- and y-coordinates, as described below.

If Other Structure, specify
If Type you selected is "Other Structure," use the box to identify the kind of structure it is. Keep your entry short (ex: Water Tower, Smokehouse, Fishing Shack).

*Role
"Role" refers to the role the Location plays in the text you are editing. Select from drop down menu either "Site of Event," when there is at least one scene in narrative that takes place at this Location, or "Only Mentioned in Text," when narrative mentions the place but never goes there for a scene. "Bundren Place" is "Site of Event" in As I Lay Dying but "Only Mentioned" in "Spotted Horses."

*Status
We decided to add this field when we noticed how many Yoknapatawpha buildings burn down. "Continuous" is the default setting, so if the Location is there when story begins and still there when it ends, you don't do anything here. But in other cases, select from pulldown menu: "Built" if Location is created during your text.
"Destroyed" if, during your text, the Location is destroyed.
"Rebuilt" if, at start of your text, there are the remains of a previous site that is then, in the course of text, rebuilt.
"Built and Destroyed" if both processes - its creation and destruction - occur inside your one text.
"Destroyed and Rebuilt" if, again, both these things happen inside your one text.

Other Texts
The other stories and novels that include scenes set at this location.
As you type a title in the window, Drupal will display list of titles with that combination of letters, and you simply click on the relevant one. This is another way program promotes consistency. If the text you want to add does not appear automatically, it means it’s not on the “Text Keys” list, so you’ll have to let me know and we’ll add it.
And how do you know what “Other Texts” should be cited here? There are good guides Faulkner’s characters and which texts they appear in, but I’m not aware of any dealing with his locations. List the texts you’re sure of, but if you aren’t sure, don’t worry – we can ask the technology to help us fill in this data later, once we’ve turned all the Yoknapatawpha fictions into data!

*Authority
I.e. the basis on which your team determined where a location goes on the map:

“Faulkner Map” = Faulkner placed it there on one of his maps (though here we will need to overwrite him at times – ex: Tull’s place in As I Lay Dying);
“Text (when unambiguous)” = the text of the story clearly specifies or implies the location;
“Context (as interpreted)” = from cues in the text of the story you can tell where the location must be, with reasonable certainty;
“Other Text(s)” = you used the way the Location is described in one or more different Yoknapatawpha fictions to help you decide where to put it on the map of your story. (Ex: in Flags there is simply no way to tell where in Mississippi is the "New Town" that Horace moves to, but in Sanctuary readers learn that it is in the Delta. Since that detail doesn't conflict with anything Flags says about the town, it is appropriate to use that information to decide where it should go on the Flags’ map. But note: if your text does provide some cues, and they are not compatible with what another text says, rely on your text for the Location.)
“Speculation” = the text gives no cues, so we had to make the call ourselves – de Spain’s plantation in “Barn Burning” is in this category. This last category exposes a major issue with “mapping Yoknapatawpha,” so we will certainly need to talk about it further. In the meantime, though, teams should use all their expertise and good judgment about locating places in this category, and “Speculation” will keep us honest for users of the project.

If Authority is "Other Text(s)," specify
If Authority you selected is “Other Texts,” identify which other texts you consulted.
Unless the Location you’re adding is in the “Out of Yoknapatawpha” category, you must supply x- and y-coordinates for it. Otherwise the technology can’t display the Location’s icon on the map.

Thanks to Robbie Bingler and the technology, getting the x and y numbers is actually pretty easy:

(1) using another browser window, go to this URL –
http://faulkner.iath.virginia.edu/prototype/?coords=y;
(2) go to "A Rose for Emily" map (by clicking on the cover);
(3) move your cursor around the map, and note the pair of numbers below and to the right of the map, just above the "Speed" sliders. These are the co-ordinates of the spots your cursor is moving through, so
(4) When you have your cursor where you want to put the Location, note the numbers in that pair. The first is the X coordinate, the second the Y, so enter first number here, and second in the next blank.

One thing to keep in mind: icons are about 30 pixels wide, so if you are putting Locations close together (ex: Varner’s Store and Mrs. Littlejohn’s in “Spotted Horses”), keep at least +/- 30 between the coordinates. (i.e. 1720, 1225 and 1721, 1230 will put the icons on top of each other).

Another thing to keep in mind: giving same y-coordinates to two Locations will align them horizontally, and same x-coordinates will align them vertically – this will let you put stores around the Courthouse Square in a line, for example.

Y
See X.

*Description
When users put their mouse over a Location icon, the Display Label you created will appear. The Description is what they’ll see when they click on the icon. Entries here should short, but long enough to allow you to provide a kind of picture of the setting. I think quoting brief passages from the text’s description of the place is a good idea too (remember: provide page numbers of quotations in parentheses).

200 words, more or less, fits into the box users will see, but if you need more, users can scroll down.

One issue that came up with first round of data entering: editors often focused these Descriptions on “what happens” at a particular Location. But that is the kind of information that will be entered in the Events category. Here, keep focus on what Location is like, on whether it appears often in Faulkner’s fiction, etc.

Once you’ve filled in a Description, your Locations work is done. Please ignore the stuff below the Description box.

---

Instructions for Characters Data Entry

Who is a character?

The original editorial teams and I agreed that our database can and should contain every person or identifiable group of people mentioned in the fictions (but see "EXCEPTION" below). This is a lot more characters than one finds in the typical print guide to Faulkner’s characters, where usually only the named people are included. Including only named figures, however, often perpetuates class and race biases – with Flags, for ex, the peripheral Baron von Rickthofen makes most glossaries (id’d as the German ace who taught the pilot who shot down Johnny Sartoris), but the Unnamed Ford Driver who caused the accident in which Old Bayard dies is never listed; Deacon Rogers is included, but not the Unnamed Blind Negro Musician who begs outside his restaurant.

At the July 2013 meeting we agreed to add "groups" to the categories of Gender, Race and Class. So Character entries now allow you to identify a Group (Individual is the default); when you select the Group option, the icon will be display the appropriate group of 3 people. As you'll see when you get to the Gender, Race and Class fields, you can also identify Groups as multi-race, -gender or -class, but when the text specifically says "a group of men" or "the black women," etc., you should still use "Male," "Female," "Free Black," etc.
So: “who is a character?” Every person or group who appears in the text. When no name is given, the Character name begins “Unnamed.”

When it’s an un-enumerated group, they will be summed up in a single entry (which also begins “Unnamed ______”) and represented by a single icon. Current practice is to default to “white male” for heterogeneous groups, for ex, “Townspeople” – but the kind of bias this necessity perpetuates will disappear once the full range of group options (White Men and Women, for ex) is available.

EXCEPTION: During 2014 the directors decided we should not create Character entries for personages who are named in the texts but not in any direct way part of the world of the narrative. These will mostly if not entirely be historical and literary figures used as points of reference, metaphors, or thematic counters. For example, when Horace mentions "Emma Bovary" in Sanctuary, or Quentin thinks about "Saint Francis" in The Sound and the Fury, or the narrator of Flags compares one of the MacCallums to "Cincinnatus," these people are not to be entered as Characters. Instead, use the Event Keyword field to record their presence in those various passages, following this formula: "Allusion:Name" – i.e. "Allusion:Emma Bovary," etc. By recording them in the Keywords field, we preserve the option of finding them again if it seems more appropriate later to enter them as Characters. In some cases you’ll have to make a close judgment call, but the distinction should be made along the lines of "Could this person ever enter the narrative world of the text?" On that basis, "Ulysses S. Grant" or "Jefferson Davis" or even "Babe Ruth" will usually be Characters, with their own entries, while "Alexandre Dumas" or "Semiramis" are not.

Note on other resources: you should generate your own list of Characters directly from the text, but as a next step some of us have found it really useful to consult the print guides to Faulkner’s fiction to double check on our lists. These include Faulkner’s World, by Thomas E. Connolly; William Faulkner’s Characters, by Thomas E. Dasher; Faulkner’s People, by Robert W. Kirk; A Critical Companion to William Faulkner, by Fargnoli, Golay and Hamblin; William Faulkner: The Yoknapatawpha Country, by Cleanth Brooks. (Brooks’ index is especially helpful when filling in the “Other Texts” field below.) If you know of other good guides to Character data, please feel free to share it with the whole group.

And when finished, the DY site will contain a complete bibliography of the works we used to create it, so if you use any other indexes, please keep track of them.

As with Locations, you will “Add a new character” for every character who appears in the narrative of your story. If you came up with a list of, say, 23 characters, you will “Add a new character” 23 times. But NOTE: You are entering data for the character as he/she appears in your text. I.e. “Colonel John Sartoris” is very different in “Barn Burning” and in The Unvanquished.

This gets complicated, and is an example of how turning Faulkner’s prose into data can be frustrating. In the “Biography” you’ll write, you can indicate something about a character’s place/identity in the larger Yoknapatawpha saga – i.e. you can say that “Colonel Sartoris” is a major figure in the whole canon, etc. But since he is only mentioned in “Barn Burning,” he’s a “Peripheral” character, not a “Major” one (though I’d still classify his “Vitality” as “Dead-ghost”) and the Biography for him you’ll write in the box on the form should focus on his “role” in this particular text. The same goes for all the other pieces of data. For example, under “Family”: Elnora in Flags is only identified as Simon’s daughter, so a "Strother," but in “There Was a Queen” readers are told she is the illegitimate daughter of John Sartoris; which Family you choose to put her in depends on which text you’re editing. Once we have all the Yoknapatawpha fictions in the site (!) we’ll create “Master Bio’s” and “Master DataEntries” for the recurring characters, but that’s not what we’re doing now. For now, base your data on the text you’re editing.

*Asterisks mean you must enter data for this field.

*Text
The story you’re editing. Choose it from the pull down menu.

*Display name
This is what users will see when they hover their mouses over a character icon, so use your judgment about what is best. One could write, for example, Aunt Jenny Du Pre, or Virginia Du Pre, or just Aunt Jenny – use the name that not only identifies this one character, but is also the name users will be most likely to recognize. (The other possible versions of a character’s name will also get recorded, see AKA below.)

*Sort name
The alphabetical, i.e. last name first, version of the Display Name – ex: "Du Pre, Aunt Jenny"

But with Unnamed characters, keep "Unnamed" first. Ex: "Unnamed Jeweler" is still "Unnamed Jeweler," not "Jeweler, Unnamed."

AKA
All the other names by which the character is referred to in your text, and if there’s a way most Faulknerians would refer to this figure that isn’t in your particular text, list that too. Ex: Bayard Sartoris is called “Colonel Sartoris” in “A Rose for Emily,” even though he never held any
military rank. His Display Name could be “Colonel Sartoris” or “Colonel (Bayard) Sartoris” or whatever your team thinks is best, but make sure to list “Bayard Sartoris” and “Old Bayard” in his AKA. Maybe even “Bayard Sartoris II.” Since the point of this field is to make sure that a search for, say, “Benbow Sartoris” throughout the Yoknapatawpha fictions finds him as "Bory” too, listing more names rather than fewer is appropriate here.

*Individual or Group
Use pull down menu.
"Individual" is the default setting, so if you're entering data on a person, you don't have to do anything here. But if you're entering data for a "Group" -- more than one person -- choose that option, and make sure you keep Race, Gender and Class choices below consistent with the composition of the Group you're entering.

*Race
Use pull down menu.
Pretty simple, until we get to Joe Christmas. Two un-intuitive details:
1. For Individual “mixed race” characters we now have 4 different sets of icons: choose "Mixed" for a character whose ancestors and black and white (mulatto, quadroon, octoroon, etc.); "MixedBlackIndian" and "MixedIndianWhite" should explain themselves; as far as I know, the "MixedIndianWhiteBlack" only applies to Sam Fathers, but in any case choose that category for characters whose ancestors belong to all three races. "Multiracial Group" means the Group you're working on contains people from more than one race.
2. Use "Indeterminable" only when you feel you must, i.e. when you don’t feel there is any basis for identifying the Character’s race, or when (as in the cases of Joe Christmas and his biological father) Faulkner deliberately makes this indeterminable. But I would say, use this option sparingly. When there is a strong presumption of race (for ex, all store owners I can think of would be “White,” even in that adjective never appears in their description, and for me all cooks would be “Black,” unless otherwise specified by text) please select that category.
You should go into specifics about either "Mixed" or "Indeterminate" in the Biography you’ll also write for each character.

*Gender
Use pull down menu.
Very simple, since I can think of no transgendered characters in Faulkner’s Yoknapatawpha fiction. Just note and use the third option ("Multigender Group") if you're entering data on a Group that includes both males and females.

*Class
Use pull down menu.
We constructed this list doing the characters in Flags, and revised it to include Indian categories and "Indeterminate" at the July meeting. At that meeting there was some resistance to this entire field; in response the next category (Socioeconomic Standing) was added. But even if you have reservations, you have to decide, based on the way Faulkner's world behaves, which of the following labels best fits each character:
"Upper Class" is for the “old families” (no matter how much money they had in the past, or have at the time of your text; Emily Grierson, for ex, belongs here), and large land and/or slave owners (ex: Compsons and Sutpens are both Upper Class).
"Middle Class” will mostly be doctors and lawyers and business owners, and their spouses and children. Most "Middle Class" people, I think, will live in town.
"Lower Class” are whites who work for the first two groups, or farm small farms they own.
"Poor White” characters are typically share-croppers or the hangers on of plantations (I’m thinking of Wash Jones and his granddaughter).
"Yeoman" we got from the older generation of Faulkner critics, mostly Cleanth Brooks. It did seem useful for people like the MacCallums, but I’m not sure how often it will be used elsewhere.
All slaves are “Enslaved Blacks,”
And at least for now, all post-emancipation blacks are “Free Blacks.”
"Indian Chief,” when the text identifies a character that way.
"Indian Tribal Leader,” for Native American characters between the rank of Chief and regular members of the tribe.
"Indian Tribal Member,” for the remaining Native American characters.
"Multiclass Group” is appropriate when you are entering data on a Group and it is not clear or clearly implied that the Group consists of people from a particular class. For example, "Salesman" (or "Drummers") would probably all be LowerClass, but "Townspeople” would be a Multiclass Group.
"Indeterminable” - another category to use sparingly, but use it when you can find no possible way to assign one of the other labels to a character or group. (The example that came up in our meeting was "UnionSoldier” or "UnionSoldiers” – though I think we can assume that when a Union
The Digital Yoknapatawpha Project

soldier is identified as an officer, the right choice is "MiddleClass" or even, if he seems sufficiently well-bred, "Upper Class," and when not an officer, "LowerClass."

Again: we can change the options later, and revise the data when we do. Feel free to use the Notes section of this page to note places where you felt the need for better or more categories.

How to handle characters whose Class changes within a text - for ex: Sutpen in Absalom! – is a question we still need to answer. For now, base your decision on how the narrator/larger narrative treats the character, and use "Notes" to call attention to the problem in each case. You can also propose solutions!

Socioeconomic Standing

The intent of this field is to solve some of the frustrations we felt about the simplications imposed by "Class" labels. You won't enter data here for most characters, but when you feel you can and should supplement the Class you chose with information about a Character's property, wealth or status, this provides the means to do that. For ex, while Emily Grierson is "Upper Class," she is also "poor." You can click as many boxes in this field as you find appropriate.

*Rank

Use pull down menu.

How important is this character in the one story or novel you're editing?

This is here partly because it seems like a significant distinction in a lot of criticism, and partly to allow users to control how many character icons appear on the map at a time.

Major = a character who plays a very important role in your text.

Secondary = usually a character who has lines of dialogue, takes actions that narrative records, etc, but not Major.

Minor = appears in scenes, but seldom or never speaks, never acts in ways that are significant to story, etc.

Peripheral = only referred to, but does not directly appear in narrative.

Ex: I’d say “Colonel Sartoris” is a Major character in The Unvanquished, a Minor one in The Sound and the Fury, and a Peripheral one in “Barn Burning.” There is lots of possible blurring here, but as long as your editorial team agrees on a rank, we will go with that.

*Vitality

Use pull down menu.

This refers to the character as he/she appears in the text you’re editing – i.e. Colonel John Sartoris “Dies” in The Unvanquished, and is a “Dead-ghost” in Flags.

I hope you all agree that the Alive-ghost and Dead-ghost options are good ones. We came up with them to indicate those characters who are not present in texts, either because (like Colonel John) they’re dead, or because (like Caddy Compson) they’ve left Yoknapatawpha before any of the narrative occurs, but while not present, they haunt the characters who are there. As the narrator says at start of Flags, the dead Colonel somehow seems more real than the two old men (Bayard and Falls) who are talking about him, in the same way that all the Compson brothers keep circling around the idea or memory of their sister. This seems like a peculiarly Faulknerian phenomenon, one that we should try to recognize in the “data.”

Most of the other categories seem self-explanatory, except perhaps "Born-and-Dies." Select this for Characters whose birth and death are both narrated in your text. Don't use this when a text simply covers a large enough chronological span for us to be able to assume someone must have been born and must have died inside it, but only when the Character's Birth-and-Death are both actually narrated in your text – for ex, Milly Jones's Unnamed Daughter in "Wash" and Absalom!

Family

Use pull down menu.

Your choices will look familiar – the Yoknapatawpha families whose genealogies have been charted by many scholars – with this exception: the black families that live with the “big” white families for generations (Struthers with Sartorises, Gibsons with Compsons) I think deserve parity with those white families, so we will include their genealogies in the project too.

Every Character has a family, of course, but here we're trying to capture the ones who are related to the Yoknapatawpha families that recur in multiple texts and generations. This piece of data will enable us, in time, to construct interactive genealogies of those families. If there are families I’ve forgotten to list, please let us know.

Occupation

Use pull down menu.
Here (with one addition: Crime) we’re using the occupation categories constructed by the Bureau of Labor Statistics (I don’t know why they left “Crime” off the list...) If you’re not sure which category to use for a specific character, go to the BLS index for help (http://www.bls.gov/ooh/). This is only for characters who get paid for what they do. Neither slaves nor, say, housewives have an Occupation. You can use “Farming,” though, if you think it’s appropriate, for farm women, and even children on small farms. This is the way users will be able to search the data for “how do the people of Yoknapatawpha make their living,” i.e. how many get it from the land, like planters and farmers, how many from professions, etc. etc. We had to come up with a short enough list to make searching possible, but there is clearly a difference between the way a Planter makes money and the way a Sharecropper does, so that’s the logic of the next item:

Specific Job
For characters who get paid for what they do: enter whatever terms your team agrees on. For example, Simon Struthers is “Domestic Service” under Occupation, but here you can specify Carriage Driver, or lots more; Dr. Peabody is “Professional” above, but here you can specify Doctor, or Physician, etc.

For characters who don’t earn any kind of salary, but whose work you want to make part of the record we’re creating: again, say what seems appropriate. “Housekeeping,” for example, or “Spouse” (I certainly think marriage to Flem is “work” for Eula).

For the sake of project consistency, please capitalize all the Key Words in your entry. Once all the teams have entered data, we’ll aggregate the terms that appear in this field and see if we can generate a list that way, or if it’s better to let each editorial team use their own words.

*First Appearance
Use pull down menu.
Here you identify the Location at which the character is first introduced into the narrative. That Location has to already exist on the Locations Key, which explains why each team should start entering data with Locations, then (2) Characters, then (3) Events...

“For characters who make it through the BLS index for help (http://www.bls.gov/ooh/). This is only for characters who get paid for what they do. Neither slaves nor, say, housewives have an Occupation. You can use “Farming,” though, if you think it’s appropriate, for farm women, and even children on small farms. This is the way users will be able to search the data for “how do the people of Yoknapatawpha make their living,” i.e. how many get it from the land, like planters and farmers, how many from professions, etc. etc. We had to come up with a short enough list to make searching possible, but there is clearly a difference between the way a Planter makes money and the way a Sharecropper does, so that’s the logic of the next item:

Specific Job
For characters who get paid for what they do: enter whatever terms your team agrees on. For example, Simon Struthers is “Domestic Service” under Occupation, but here you can specify Carriage Driver, or lots more; Dr. Peabody is “Professional” above, but here you can specify Doctor, or Physician, etc.

For characters who don’t earn any kind of salary, but whose work you want to make part of the record we’re creating: again, say what seems appropriate. “Housekeeping,” for example, or “Spouse” (I certainly think marriage to Flem is “work” for Eula).

For the sake of project consistency, please capitalize all the Key Words in your entry. Once all the teams have entered data, we’ll aggregate the terms that appear in this field and see if we can generate a list that way, or if it’s better to let each editorial team use their own words.

*First Appearance
Use pull down menu.
Here you identify the Location at which the character is first introduced into the narrative. That Location has to already exist on the Locations Key, which explains why each team should start entering data with Locations, then (2) Characters, then (3) Events...

“First introduced into the narrative” means “referred to by narrator, or other characters,” as well as “directly present in the scene.” Ex: in Flags, Johnny (Young John) Sartoris is mentioned by his brother and uncle at the Sartoris Plantation, so that is where he “first appears.” When users ask site to Display Characters on the map, the First Appearance field decides where the icon for each character will appear (unless user asks to see Characters displayed by the next field, i.e. “Home”).

Home
Use pull down menu.
Where the character lives, if known. For most characters this won’t be known, but when text does establish that, the “Home” function will allow users to ask the map to display where the various characters in a text live. Exs: Old Bayard and Simon (both of whom "first appear" in Flags at the Bank) are "at home" on the Sartoris plantation; also in Flags, Rafe MacCallum first appears in Courthouse Square, but he lives with his father and brothers at the MacCallum Place.

It’s interesting to note what kind of characters have established residences in Faulkner’s fiction.

Date of Birth
Enter data here only if text allows you to provide a birth date within a specific range. There are many possible legitimate clues – tombstones, or character’s age or stage of life is mentioned, or (for recurrent characters) birth date is mentioned in another text and seems dependable, or we’re told character was born “before the war” or “when his parents were old,” etc etc. If you feel you have a good basis for providing a DoB, please do so; I can imagine useful searches for “all Faulkner characters born like him at the end of 19th century,” etc. But for many characters there is no such textual basis, in which case you simply leave this blank.

I apologize in advance for how un-intuitive the entry system here is, but we were stuck with the options Drupal gives us.

(1) If you know someone’s precise birthdate, just enter it in the box, being sure to use the specified format: yyyy-mm-dd. (Ex: Benjy Compson was born 1895-04-07.)

But Faulkner seldom gives us a precise birthdate, or even a birth year. In that case, check “Show End Date.” This will open up an additional box, so you can provide a date range. (Note: the second box is not for a death date: “end” means “end of the range of possible birth dates.”)

(2) If you can date a birth to a particular month – say December 1910 -- range is 1910-12-01 in first box, followed by 1910-12-31 in second. This will display for users as “December 1910,” but search functions require us to enter the data in a certain way.

(3) If you can only narrow it down to a specific year, use yyyy-01-01 and yyyy-12-31. (Ex: John Sartoris was born in 1823, so in first box you enter 1823-01-01, and in second box, 1823-12-31.)
Multiple years follow same format – i.e. 1834-01-01 and 1836-12-31 would mean (and will be displayed to users as) “1834-1836.” There are obviously a lot of characters for whom you’ll need to use a very wide range, but that’s okay.

Origin
If known, where a character was born, or lived before coming to Yoknapatawpha. I’m not sure what options are best categories here, so for the time being narrow it down as much as possible – i.e. Massachusetts, if story makes the state clear; North, if that’s as specific as story gets; England, or Europe; etc., or even just “Out of Yoknapatawpha,” if that’s the most you can say. And if this is a local man or woman, someone your story makes it clear was born in Yoknapatawpha, enter “Yoknapatawpha.”
(I’ve been assuming almost all the “Negro” characters – except the slaves identified as coming with owners to Yoknapatawpha, or bought in New Orleans, etc. – are local, but this is something we can talk about.)
Put whatever place term seems most appropriate to you, and at the end of the year we’ll collect all the examples and see if we can organize them into a usable list.

For many characters we won’t have a legitimate basis for identifying a place of origin, but when we do, it’s worth recording. I can imagine a scholar wanting to know “how many Yankees appear in all Faulkner’s Yoknapatawpha fiction?” and it’s certainly interesting to note how many of the major characters in Light in August are not originally from Yoknapatawpha – so I hope we can arrive at the right way to enter all that can be known about that kind of topic.

Cause of Death
Use pull down menu.
But enter something here only if you entered "Dead," "Dies," or "Dead-ghost" under the Vitality field above. Don't enter anything for characters who die in a different text, after the one you're working on (ex: Mrs. Compson dies in the “Appendix: Compson,” but is – alas – very much alive in The Sound and the Fury).

Ontological Status
Use pull down menu.
Apologies for the name of this field. It replaces the earlier "Real?" with a more supple way to identify those Characters for whose existence it's not appropriate to assume that "Faulkner" is accountable - i.e. they appear in Yoknapatawpha, but have their origin in reality (like Babe Ruth) or a text Faulkner didn't write (like the Bible), or spring from the imaginations of characters in the text. You won't use this field very often, but when it's appropriate your choices are:
Historical/Real = Ulysses S. Grant as well as Babe Ruth, etc.
Literary/Mythic/Biblical = for a Character who exists in a non-Faulknerian text of any kind (the one clear example that occurs to me for this category is the "Prince," i.e. Satan, in The Hamlet, along with his demons; since they are given lines to speak, etc, they have to be entered as Characters, though how we'll locate their First Appearance on our map is a question that will take some creativity to answer!)
Created by a Character in Text = the easiest way to "explain" this option is probably the Lawyer in Absalom!.. Quentin and Shreve interpolate him into the Sutpen story, but there is no evidence that he exists outside their version of the past (there's a neat moment on the Faulkner at Virginia tapes when he himself is asked if this Lawyer "actually existed or Quentin and Shreve made him up," and admits his own uncertainty.
Note 1: A number of the "Historical" and "Literary" people who are named in the texts should not be entered as Characters; see the EXCEPTION described under the heading "Who is a Character?" in the Instructions for Characters. Note 2: this field should not be used for characters for whom real people have been identified as the "originals," such as Colonel Falkner/Colonel Sartoris, etc. Though in time we might very well want to create a way to include that kind of data in our system.

Narrator
Use pull down menu – IF and only if the character you’re working on is the source of some part of the text you’re working on, i.e. serves as a source of the narrative at some point, at least. In Flags, for example, the third-person narrative gives way at a couple points to a stream-of-consciousness narrative of Horace Benbow’s thoughts. But this category does not include indirect representation of a character’s thoughts; source for that would still be the third-person narrator.

Disability
Check all the boxes that apply IF and only if the character of the Character you’re working on is strongly associated with one or more disabilities. There will be many ambiguous cases here, I think - is Old Bayard in Flags deaf enough to be clinically labeled "audio impaired"? or just deaf enough not to have to hear Jennie when she scolds him? In this case I’d check that box, because the very first description of him (and Will Falls) calls attention to their "deafness," and that part of his character repeatedly comes up in the rest of the narrative. Darl in As I Lay
Dying is almost certainly "psychosocially impaired," which is the term we'll use for "mad" or "crazy," but is Vardaman too? I think the ultimate criterion we should use in deciding when to check a box here, and which to check, is our sense of whether Faulkner wants us to identify a character with a particular (set of) disability(ies).

There are ethical reasons to be uncomfortable with the kinds of judgments we have to make in the case of this field, but disabilities are so important in so many Faulkner texts, and so many readers associate his fiction with this characteristic, that we need the category. The terms we're using are ones used in contemporary disability studies. When we create a "user's manual" for the whole DY project, we'll probably want to include a kind of glossary which can connect these terms to the vocabulary Faulkner and his characters used - where Benjy, for ex, is a "natural," not a "cognitively disabled person."

Ethnicity

This too is a category saturated with sources of ambiguity and discomfort, but again, the idea of an ethnic "us" and various ethnic "thems" seems an inescapable component of Faulkner's Modernist sensibility. In a truly egalitarian cosmos, if anyone has an ethnicity, then everyone has one. But we're interested in capturing the world Faulkner's imagination created, which seems, for ex, to have more than some sympathy with Quentin when, walking through urban Boston, he thinks to himself "land of the kike home of the wop," or with Aunt Jenny when she walks into her dining room and almost instinctively recognizes that Narcissa's visitor is both a "Yankee" and a "Jew." So (as with Disability) here you should only use this category to note when a character is marked in the narrative by a marked kind of ethnic "Otherness."

There is no list of "ethnic types" for you to choose from - yet. When you want to note a particular character's ethnicity, use your own critical judgment about what word(s) to put in the box. After we've been doing this for a year or two, we'll create a list of choices from the words we've come up with together this way, and revise the entered data accordingly. (By the way, I wouldn't say "Yankee" is an ethnicity. But for the man Jenny is reacting to, I would definitely put "The North" in the "Origin" box - and "Jewish" in the "Ethnicity" one. Black or Indian racial "otherness" is not an ethnic difference either, nor would I say that in Faulkner's world Shreve's "Canadian-ness" is an ethnic category - though "Mexican" certainly could be. But we need to work this out together, so feel free to raise questions about particular cases - that's one good use for our new wiki.)

Other Texts

Try to list all the other Faulkner texts the character appears in. I use the various guides to Faulkner to help out my memory here, especially the list at the back of Cleanth Brooks’ book. But don’t spend too much time trying to make absolutely sure you haven’t overlooked anything – this is another of the fields which we can ask the technology to help populate, once we have entered all the Yoknapatawpha fiction into Drupal...

*Biography

I hope these will be fun to write. At least the category allows you to use your own words, rather than data values! This prose description is the first thing users will see whenever they click on a character icon, so write it with that rhetorical situation in mind. But also keep in mind that we can’t control the order in which they choose characters to click on, so each entry has to be complete in itself. (I.e. if you were doing “Barn Burning,” it might be confusing to call Lizzie “Lennie’s sister,” since a user won’t necessarily have clicked on Lennie first; it would be clearer to say “Lennie Snopes’ sister” or perhaps “Flem Snopes’ sister-in-law and Sarty’s aunt”)

At the same time, entries should be concise (the popup box displays about 200 words, though users can scroll to read a longer passage, if you feel you need to say more than the box will hold).

The Biography should focus on the character as he or she appears in the text you’re editing, but this is a place where you can also indicate how character shows up in other fictions. For example, a bio of Jenny Du Pre in Flags could mention that she is the “queen” in the short story “There Was a Queen,” or that Flem Snopes is “one of the central characters in the larger Yoknapatawpha saga,” etc. But the Biography of Flem in a particular story is not the place to go into detail about who “the Snopeses” are, or what that family represents in the larger canon, unless that issue is a major theme in the story you’re editing.

You don’t have to list here all the other texts in which a character appears, though, because users will see that data via the previous category. As with Locations, in the first round of data entering, editors often focused these Biographies on narrative more than character, on what a character does in the story rather than who he/she is. Summing up a character’s role is fine, but going into detail about actions isn’t appropriate here; that kind of information will be entered in the Events category. Here, try to keep focus on who Characters are.

Please ignore the “Edit Summary” link. That’s a feature of Drupal we can’t disable, but we’re not using.

Notes
All the data you’ve entered in the various fields above will, I have no doubt, be useful to future users of our project. But as I said in the general comment on Characters, there’s still a lot of creative thinking that we need to do about what other kinds of data we should try to capture for Faulkner’s characters. So I want you to use this “Notes” field as a kind of scratch pad. Think about what else those future users – whether they are high school teachers, college students, dissertating graduate students, publishing scholars in American studies, English, history, etc – might want to search for, and also what the particular character you’re working on now brings to mind, and make all the comments you can or want to about what is not captured by the existing data fields. Examples could include such traits as “religiosity” or “education,” etc.

My thinking on this has gotten me as far as wanting to create a list of key words or phrases, which might provide basis for a new data field, or, conversely, we might want users to have access to a lookup table of such key words that they can search by. In either case, I suggest we start by using this “Notes” box at least to enter any and all searchable terms that come to mind and seem significant enough to keep thinking about. For example, Benjy Compson’s “Notes” section might include:

“organizing point of view, handicapped (mentally retarded, mute), wounded (castrated), childlike, Christian symbolism (33 years old), renamed,” etc etc

If all of us spend some time generating such lists for each character in the story, we can aggregate the results and have a great basis for improving Character data by leaps and bounds.

Once you’ve made any “Notes” you want, your Characters work is done (at least for the time being!). You can ignore everything on this page below "Notes."

---

Instructions for Events Data Entry

What is an event?

First of all, our goal here is not to summarize or paraphrase a story’s plot, The first groups of editors felt a strong pull in this direction. However, we don’t want students to see our project as a substitute for reading Faulkner, a kind of high-tech cliff notes. Our goal is to give users new modes of appreciating, exploring, interrogating Faulkner as they read the texts.

So “Event” is not “Plot summary.” But among those first groups of editors there was quite a bit of disagreement about what else it might be, leading to a lot of inconsistency. We used the Faulkner&Yoknapatawpha symposium in July 2012 as an occasion to discuss this issue, and reached came up with the consensus definition you'll read here. We can always talk in the future about revising the definition, but until we reach a new consensus together, please abide by the current one.

At the fundamental level: any continuous scene in a text is one Event, no matter how many pages long it is. But “continuous” is not a self-evident term. I’d say 1 setting, 1 unbroken length of time, 1 main focus and 1 narrative style is 1 continuous Event. When we did the data for Flags, for example, we decided the entire week that Bayard spends at the MacCallums’ is 1 Event, but others could decide “a new day” in such a term. I’d say 1 setting, 1 unbroken length of time, 1 main focus and 1 narrative style is 1 continuous Event. When we did the data for Flags, for example, we decided the entire week that Bayard spends at the MacCallums’ is 1 Event, but others could decide “a new day” in such a

theory. The first event of the text is the events that occur during the first period of time. Each event is a separate and distinct event. When we did the data for Flags, for example, we decided the entire week that Bayard spends at the MacCallums’ is 1 Event, but others could decide “a new day” in such a case makes a new Event.

There are many potential ways to define “not-continuous.” We agreed that shifts in time should be noted as new Events – i.e. in Flags, when after the narrator has described Sartoris family sitting together at Christmas, 1869, Aunt Jenny begins telling the story of how Carolina Bayard was shot in 1862, a new Event (with a new Location, Date, etc) begins. Since dramatizing the interpenetration of past with present is one of Faulkner’s most recognizable preoccupations, probably every time shift should be flagged as a separate Event – i.e. not just when a character tells a story about the past, but when a character telling about a scene in the past is interrupted by someone in the present, then goes back to the story after the interruption (that would be two separate Events - the interruption and the resumption of the story), or whenever a character has a memory, no matter how brief a space on the page it occupies. Or, in some cases, arrival or departure of a specific character at or from a scene might need to be flagged as “different Event.”

Similarly, changes in narrative technique can also be flagged at new Events, as when in Flags, the third-person omniscient is disrupted by those sudden “stream-of-consciousness” moments giving reader direct access to Horace’s thoughts.

The number of Locations we can have for any one text is limited by the fact that all Locations show up as icons on our maps, and too many too close together will become unusable. But theoretically there is no limit to how finely you can break a text into separate Events. Practically, on the other hand, there is the amount of work involved in creating and entering each Event - which of course you’ll have to do. And, Events also display on the maps, as those splashes that fade as new Events occur, so there is a reason not to multiply the number of them indefinitely – i.e. we don’t want to visually overwhelm users. That is something to consider as you decide on how many Events your story should be broken into, but it is not the only thing to consider. For now you should use your judgment as a team, come up with a rationale that you can share with the rest of us at some point in the future, and then make sure you apply those criteria consistently throughout your text.
NOTE: You should enter Events data only after you’ve done both the Locations and Character databases, because some of the entries here have to draw on those.

*Asterisks mean you must enter data for this field.

*Text
Use pull down menu. Select the story you’re editing.

*Page Number
The page number on which the scene/episode/memory begins.
Reminder: use Collected Stories or Uncollected Stories or the Vintage International paperback editions of the novels ("Corrected by Noel Polk") as source texts. Ex: first page of “Spotted Horses” is 165 (from Uncollected Stories).

*Order within page
Note that default setting is “1,” so if no other Event begins on the same page as the Event you’re adding, you don’t do a thing here. But if 2 or more Events begin on this one page, you must enumerate them as 1, 2, 3 etc, in the order they appear. This enables the animation to play the first event on page X before the second event of that page, etc.

*First 8-10 words of event
This provides users with a different edition with a way to identify the Events. Don’t use quotation marks, unless the words you’re entering are in quotation marks in text.

*Page Event Ends
I think this is self-explanatory, but I’ll just add: this page number should line up with “page next event begins” (i.e. don’t begin an Event inside the page numbers of another one).

*Location
Use pull down menu. Select the location of this Event from the LocationsKey. This is one reason why you should enter Locations and Character data before doing Events. If you haven’t already created the Location, you won’t be able to complete the Event entry.

*Characters Present
List here every character who is physically present within this Event, whether they speak or explicitly act. Even if they are not explicitly named in the passage, but you have no doubt they are "present" (as the Bundrens, for ex, are together on the wagon carrying Addie's body), list them. But note: if Jenny is talking in 1869 to 3 Sartorises in Mississippi then begins telling a story about Carolina Bayard getting killed in the Civil War in Virginia, the Mississippi Sartorises are not "present" in the Event of the 1862 story, only the people on that Virginia battleground. Each character gets his/her own box; use “Add another item” for each additional character. For each character, type in by the DisplayName you entered on Character data form; as you type, you should see the correct name displayed as a choice. This will give you a way to check for completeness and consistency too – if you don’t see the name you’re trying to enter appear, either (1) you haven’t yet created the entry for this Character, so stop and do that now, or (2) you typed the name differently when you created the Character data. (And please just ignore “show row weights” – I have no idea what it means, but Drupal won’t let us erase it.)

*Characters Mentioned
List here the characters who are referred to within the Event, but not physically present (ex: “Colonel Sartoris” is mentioned in the legal scene at start of “Barn Burning,” so is “Mentioned” in the Event). You will enter each Character Mentioned one at a time, as in the previous field.

*Narrative Status
Who/what is responsible as the source of this Event? “Narrated” = a third- or first-person narrator. (Ex: when Bayard describes Col. Sartoris escaping from the Yankees in The Unvanquished, it is narrated.)
“Told” = a character in the novel or story is telling the event to other(s). (Ex: when Old Man Falls describes the same escape in Flags in the Dust, it is told. You can indicate who is "telling" these kinds of Events by beginning your Summary of the Event with a phrase like "Old Man Falls tells the story of . . . " or "Will Falls tells Bayard about . . . " etc.)

“Remembered” = readers are seeing the event as a character is remembering it in his/her mind, whether you think the event “really” happened or not. (Ex: Quentin’s conversations with his father in The Sound and the Fury are remembered.) You can indicate who is doing the remembering by beginning your Summary with something like "Quentin remembers . . . ."

“Hypothesized” = a character or characters are constructing a possible version of what happened. I don’t like this label (if you think of a better one, please let me know), but we need a category for, for example, all that speculative narrative reconstruction that goes on in Absalom! You can indicate who is responsible for this possible Event by beginning Summary with something like "Quentin and Shreve speculate" or "theorize" or "hypothesize," etc.

"Narrated+Consciousness" = Choose this for those times when Faulkner combines third-person and stream-of-consciousness narrative techniques in a single paragraph. I think most such Events occur in Light in August as in the novel’s very first paragraph: "Sitting beside the road...Lena thinks, 'I have come from Alabama...' Thinking although I have not been... " (By the way, with this kind of Event I'd include "Style:Italics" and "Style:Present_Tense" in the Keywords.) But there are several such places, where omniscient narrative gives way to the direct representation of a character's thoughts, in Flags as well. (Stream-of-consciousness narrative, i.e. the first 3 sections of Sound and Fury or all the chapters of As I Lay Dying should be identified as Narrated, unless a character is Remember-ing an earlier event inside the stream-of-consciousness passage - who said Faulkner was easy?)

*Date
This is the same awkward procedure as assigning DateOfBirth on the Character data, and again I apologize for how un-intuitive the entry system here is. (The future scholars who search our data, though, will bless your work!)
If you know the date of an event precisely, just enter it in the box, being sure to use the specified format. I.e. the date of the event “Sutpen and Henry quarrel about Bon and Henry renounces his birthright” goes in the box you see on the form, as 12-24-1860.
If you do not know the specific day, check “Show End Date,” which will open up an additional box, so you can provide a range within which the event fell.
If you can date it to a particular month, say December 1909 when Quentin comes home for Christmas, range is 12-01-1909 in first box, followed by 12-31-1909 in second. This will display for users as “December 1909,” but unfortunately you cannot simply enter the data that way.
If you can only narrow it down to a specific year, use 01-01-xxxx and 12-31-xxxx. Multiple years follow same format – i.e. 01-01-1834 and 12-31-1836 would mean (and will be displayed to users as) “1834-1836.”
This is one of those places where we have to submit to the technology, but in the long term will mean someone could ask our project to show “everything that ever happens in Yoknapatawpha in chronological order,” which I for one am looking forward to seeing!

Chronological Order
Don’t start entering this until after you’ve created all the Events in your text. Once you’ve done that, you go through them and identify the order in which they happen chronologically (as opposed to the order in which they appear narratively; i.e. date of event vs page number as ordering principle). For ex: in The Sound and the Fury Event#1, chronologically, is probably the scene where Benjy remembers being down at the branch the day Damuddy died. Event#2 is the next scene from that afternoon/evening, etc.

Earliest Event is number 1, next is 2, etc. Start with whole numbers, but we’ve set Drupal up to read decimal intervals (i.e. 1.1, 1.11, 1.2, etc), so if you leave one or more out the first time you won’t have to renumber the whole list – if it occurs chronologically between your 4th and 5th Events, you can identify it as 4.5, etc.

Drupal can help (up to a point) with this process. If you (1) go to Events, (2) use the Apply button to select just the Events in your text, then (3) click on the Date category, the Events will be resorted according to the dates you assigned them, oldest to newest. But you’ll still have to decide the exact order yourself. In some cases you may have to rely on your informed judgment to decide which of several parallel Events comes earlier, but that’s okay.

You’ll be able to check to see that you included all your Events by looking at the Chronological Order column on the Drupal-Events page.

*Era
Use pull down menu, where you’ll find the chronological parameters for each Era.
If you’re not certain, pick the most likely time frame.
This piece of data will allow users to ask the site to display, for example, “everything in all the Yoknapatawpha fictions that occurs during the Civil War” – which I am also looking forward to seeing.

*Summary*
Here you'll write in a brief synopsis of "what happens" in the Event. Remember that we don't want our project to provide students with a kind of electronic cliff's notes, i.e. Summaries should not provide substitutes for reading Faulkner's text. Keep them short, just long enough to jog the memory of someone who has read the text.

*Keywords*
We're going to start with a text box here. Each team will fill it with all the relevant Keywords they agree on for each Event. Once we've seen what teams come up with, we'll turn the words into a "controlled vocabulary" so that we can be consistent about terms across texts and groups. A "controlled" list means that at some point in the future you'll only be able to type in words that are in that vocabulary. (We'll also be able to expand and revise the vocabulary as the project moves forward, and we will certainly want to.)

Attaching useful and consistent Keywords to each Event is going to require a lot of work. But in the project's long run, it may turn out to be the most valuable contribution DY makes to Faulkner studies. If we can create a rich and supple enough database, scholars and students will be able to query the entire Yoknapatawpha at one time: for ex, a scholar writing on an issue like "racial violence" or "environmental degradation" in Faulkner could use Keywords like "violence" and "race," "land use" and "farming," to locate all the scenes in the stories and novels that might be relevant to such analyses. Keywords should be conceived in terms of both "what goes on in Faulkner's fiction?" and "what kinds of questions will our users want to ask our database?"

While you're thinking about what Keywords you want to create for each Event, it might help to break the possibilities down into subsets, so here's a potential list of categories of Event Keywords, with just a few examples of each:

Environment (twilight, drought, kitchen, Civil War, woods, urban.....)

Actions (drinking, cooking, eating, moving, fighting, sex.....)

Emotions (anxiety, desire, fear, shame, rage.....)

Cultural issues (religion, race, gender, slavery.....)

Relations (family, father-and-son, fellow-workers, fellow-congregants.....)

Themes and motifs (time, past-in-present, violence, return-from-war, meaning, honor.....)

Aesthetics (italics, long sentences, Biblical or literary allusions.....)

Other (anything that you think is important to have, but doesn't seem to fit into any of these subsets; for example, here's where we could begin creating a database of animals in WF, so you could say "dog," or "deer"; or objects ["family heirloom"], etc. SPECIAL NOTE: If the Event is primarily or powerfully organized as a form of movement, use "Movement" as one of its Keywords, and add the means of moving, i.e. "Movement-wagon," "Movement-foot," "Movement-car," etc. In time we will try to animate movement on the maps, and this keyword will help us identify each instance of it.

Two issues about Keywords that remain unresolved: What is the appropriate level of granularity we should aspire to? I.e. do we want to say "drought," or "weather"? "Civil War" or "war"? I think we should always strive to walk an invisible line between (1) how much detail it would be great to let users choose from, and (2) the realistic limits on the amount of time and energy each of us can spend entering data for this project. Rather than trying to work this out in advance, each team should create Keywords according to whatever standard feels right to you – once we have the lists, we'll see how much revision we need to do to get some kind of consistency across teams and texts.

Do we want to use categories like "Environment" or "Aesthetics" in the public version of this? i.e. should our eventual user community (students, teachers, scholars and critics) be able to SEARCH by these subsets of keywords? My own feeling is that we won't need to, and shouldn't; that it will be enough to give users the complete list of Keywords they can use as search terms. But we can discuss other possibilities later. In the meantime, I'm simply providing the subsets now as a way to encourage teams to think in different directions about what might be included in the keywords you generate for each event.
And from the lists we've generated so far, one issue that's going to require a lot of work by all of us is consistency. The power of this resource depends on consistent aggregation across all the texts. If only one team, for ex, enters animal keywords, a search for, say, "animals" or "horses" will not only be a lot less valuable; it will be extremely misleading, because it will seem that "horses" only appear in that one text. I know that in practice aggregation will often mean aggravation, or at least a lot of work. But one way to work toward our goal is for each team to look at the keywords in other project texts as you develop the keywords for your Events.

*Notes

Use this field like a scratch pad, or the blank canvas of the conceptual future of our project. Use it to note anything about an Event, or Events in general, that you wanted to enter into the database but couldn't.

Please ignore the “Edit Summary” link. That's a feature of Drupal we can’t disable, but we’re not using.
During the weeks of April 14 – May 3, I was fortunate to spend my MLIS practicum at MITH under the supervision of Trevor Munoz. My main reason for applying to MITH as a potential practicum site was my background in the digital humanities and library and information studies, as well as my interest in data curation and research data management in the humanities. I also wanted to apply the skills I learned in the “Data Curation for Digital Humanists” course at the Digital Humanities Winter Institute hosted by MITH and the University of Maryland in January of 2013 (taught by Trevor and Dorothea Salo from UW-Madison.) I was pleased, then, to discover what a unique project Trevor chose for us during my time at MITH.

Our main project was to imagine ways to curate the publicly available data from the New York Public Library’s “What’s on the Menu?” (WOTM) special collection of digitized historical menus. The Library crowdsourced the transcription of the digitized menus via the project website and compiled the completed menu transcriptions into large spreadsheets, which are updated every two weeks as more data continue to be produced. Recently, NYPL Labs added a geo-tagging tool to add geospatial information to the historical menus, where available. An award-winning digital project, NYPL’s “What’s on the Menu?” is a great resource for exploring data curation in practice as it is free, openly accessible, and a subject of interest to humanists.

Our task as data curators, therefore, involved cleaning up, organizing, classifying, describing, structuring, representing and otherwise making more accessible the data provided on the “What's on the Menu?” site. A user exploring the project site will quickly discover that, while over 26,000 menus have been digitized and displayed online, one can only browse them by the year the menus were produced, the name of the restaurant that produced the menu, and the total number of dishes that appeared on the menu. In other
words, there is no thematic or categorical classification of these menus that would appeal to the discovery of the rich historical and cultural information contained in this unique collection. We decided that categorizing and classifying the WOTM data to support richer browsing was curatorial activity that might add value.

Before we began to categorize and curate this data, however, it was important to take stock of our data at hand. Essentially, we downloaded two spreadsheets from the WOTM site: one “Menus” set containing over 28,000 rows of data (the names and types of menu pages) and one “Dishes” set containing over 400,000 rows of data (the individual names of dishes that appeared on those menus, including their prices). After assessing the data, we wrote a data management plan (in best Digital Humanist practice) for the work we were about to begin in curating the Menus and the Dishes data.

Use Cases

While we could speculate on the potential usefulness of the data in this collection, it’s always better to have actual evidence of what users want. NYPL Labs graciously shared summaries of the requests they have received from people requesting credentials to use the WOTM application programming interface (API). These short description served as evidence of user needs, which would help guide our decision-making process. Historians, social scientists, journalists, literary food scholars, chefs, novelists, teachers and students, as well as general enthusiasts all showed interest in access to this data. In order to make the menus more accessible for these potential users, we had to imagine what types of questions they might ask of this information. This would inform the general working taxonomy, or information architecture, of our data set.

Data Cleaning

First, however, we needed to clean it up. As you can imagine, crowdsourced data tends to me messy, containing many spelling variations, typos, ambiguities, and missing fields. We used OpenRefine (formerly GoogleRefine) software to cluster and rename data fields that we considered of particular use or interest to future users of this collection, principally names of the businesses offering these menus and also, where present, the names of the categories (supplied by original cataloguers?) to which the menus had been assigned. While OpenRefine helped with the initial clustering, the large number of name and spelling variations meant some tedious line-by-line editing. This is also the data curator’s job.

Categorization

Upon a general data cleanup, I was excited to launch into the categorizing! While working on my taxonomy of menus and dishes, I learned the difference between taxonomies, thesauri and ontologies, and have built up a nice working list of controlled vocabularies to consult for various subjects. Overall, when creating the taxonomy of the menus, I focused on what the potential users of this data might seek in this unique collection. As such, I developed
three levels of classification with four broad categories by which to group the menus. These included the *Hosting Organization* that sponsored the menu, *Type of Meal* the menu reflected, the *Restaurant Address* where the meal was held, as well as the *Type of Gathering* for which the menu was produced. For example, we decided that users might be interested in exploring political and military meals held in the “What’s on the Menu?” collection, as well as browsing the menus created for special occasions, such as George Washington's Birthday meals. While our taxonomy structure is not the definitive version of this data, Trevor and I nonetheless believe that providing users with the ability to browse all wedding menus, High Society Banquets or meals held for royal individuals, for example, was a value-added service that data curators can provide. In other words, our categories were additional ways to enter, explore and understand this special collection that lead to discovery and learning.

User testing and evaluation was another important step in the curation process, as it allowed us to see whether we were on the right track with our curatorial practice. While we may have been following good metadata standards, referring to controlled vocabularies and linking URIs to our classification terms, none of this work would ultimately have mattered if the users could not navigate the data quickly and easily. I discovered a useful tool called Treejack, which allows information architects to test their tree structure (the organization of information) by providing anonymous users with several tasks to complete. We also asked our test subjects to answer three brief survey questions about the choice of terms used to create the categories for the menus. Based on the feedback we received, we changed the placement and labeling of certain categories. Ultimately, a final user testing of this data curation project would help evaluate it as a whole.

**Prototyping**

Finally, in my last week, we decided to develop a kind of proof-of-concept for our data curation work and display it online. Inspired by Aaron Straup Cope’s talk at the Library of Congress about his work on “Parallel Flickr”, Trevor and I tried to imagine our own “Parallel Menus” to experiment with our newly-minted categories for this data. Initially, after inserting the two spreadsheets into a MySQL database hosted on my University of Alberta web server, I created a query to display each category on its own static web page by writing some PHP scripts. Trevor, however, taught me about Bootstrap, a front-end design framework, that allowed us to make our Parallel Menus a little more interactive through Java Script. Finally, using the API key which Trevor obtained from NYPL, we were able to get thumbnail images of our menus from the “What's on the Menu?” API service and insert them into our web code to entice the users of the taxonomy into browsing our site.

**Next Steps**

Clearly, my three weeks at MITH went by quickly, and I didn’t have a chance to work on the Dishes side of our WOTM data. Nonetheless, I kept documentation for my work as we went along, and even developed some categories and controlled vocabulary terms for organizing this much larger
data set. Trevor will be teaching several Data Curation workshops in the coming months, and I hope that he will be able to expand on my practicum work by getting students to work on the Dishes part of this project. I hope that the “What’s on the Menu?” Data Curation project will eventually be handed over to NYPL to help make their site even more usable to the many visitors they receive each day. I would be honored if, upon handing it back to the New York Public Library, they used any of our suggestions with regard to organizing and classifying this data.

During my practicum, I appreciated participating both in the administrative duties of digital humanists, such as attending project meetings, policy and grant writing, and data audits as well as in more creative tasks such as attending talks at the Library of Congress, exploring new tools, and participating in the MITH Incubator project by helping librarians develop their own research projects. I especially valued the way Trevor combined the practical with the theoretical, getting me to think about the broader implications of the data curation process – its costs, its biases and limitations, its potential, its role in the creation of new knowledge. Working on this project gave me a whole new perspective on research data in the humanities, and I believe it will help me complete my thesis, as well as further my career. This was definitely a transformative experience.

Overall, I am incredibly grateful to have had the opportunity to put my skills and interests in digital humanities data curation at MITH with Trevor’s leadership. In addition to meeting the wonderful MITH team, many visiting scholars and the larger DC digital humanities community, I learned many exciting and useful things in my time at UMD: APIs, JQuery, LCSH Linked Data Service, OpenRefine and its RDF plug-in, TreeJack and tree testing, GitHub, Freebase, along with many others. I look forward to visiting MITH again in the future and launching DH curation projects of my own!

By Lydia Zvyagintseva | Fri, Jun 21, 2013 | Community

One Comment

Mauricio Giraldo June 21, 2013 at 12:05 pm - Reply

the menus geotagger has been fixed! (sorry for that)

Leave A Comment
What IS on the menu?
More work with NYPL's open data, Part One

Since I started teaching short courses on humanities data curation on semi-regular basis (first as part of MITH’s digital humanities training institute and then as part of the Digital Humanities Data Curation institute), I’ve been looking around for suitable hands-on exercises to help people “get a feel for” different aspects of the work involved in curating data in a humanities context. Maintaining the usefulness of data to researchers can involve planning, describing, building collections, and even tasks that shade into digital
preservation like migrating data to new media. Curation can also involve “cleaning,” normalizing, reconciling—what we might call “munging” data—probably (hopefully?) for the purpose of creating better search, retrieval, or indexing. The open data generated by the New York Public Library’s What’s on the Menu? project has been a great testbed for experimenting with these latter kinds of practical curation work.

Lydia Zvyagintseva, a Master’s student from the University of Alberta visiting MITH for a practicum, did great work exploring possibilities for how additional, curator-generated facets for browsing data about events and locations could add research value. In the most recent Digital Humanities Data Curation workshop (which I am fortunate to co-teach with Dorothea Salo and Julia Flanders), we worked through exercises on significant properties and potential user needs related to the menus data. In this post, I’ll describe
some more exploratory work
I’ve done since the most
recent workshop.

Beyond Access and
Preservation

What’s on the menu? is a
useful curation testbed
because the New York Public
Library (NYPL) has already
done a generally excellent
job providing access to the
data. The menu transcription
project has been running
since 2011, and it has been a
wild success. Volunteers have
used the site to transcribe
almost 17,000 digitized
historic menus (as of the time
of this post). NYPL makes all
the menu data available for
bulk download and also
provides an application
programming interface (API).

Front page of What’s on the Menu? (as of August
2013).
Beyond this impressionistic sense of “good access” to the data, we can evaluate the NYPL’s arrangements for access according to a commonly-used quality measure like Tim Berners-Lee’s 5 Star Linked Open Data scale. The *What’s on the menu?* data set scores fairly well by this measure—somewhere between 3 and 4 stars on the 5-star scale. The data is available on the web in a machine-readable, structured, non-proprietary format (stars 1-3). The criteria for the first star also specifies that data should be distributed with an open license and here we could quibble a little with the existing provisions for access. The “Data” page at the *What’s on the menu?* site states that there are “No known copyright restrictions on this material” but asks those who use the data to
“credit The New York Public Library as source on any applications or publications.” The phrase “no known copyright restrictions” echoes the language of the Public Domain Mark suggested by Creative Commons but it’s not entirely clear that NYPL’s intent with this data is wholly the same as that underlying the Public Domain Mark. Perhaps formally using the Public Domain Mark would help clarify that this is truly open data? (I offer this suggestion tentatively because I know that NYPL has some very good copyright advisors and so on the whole, I think we can give What’s on the menu at least 3 open data stars.) NYPL also uses HTTP URIs as identifiers for things, which is part of the criteria for 4-star linked open data, but the data is returned via the API in either JSON, or (a custom) XML rather than using the most-W3C-blessed standards (RDF and SPARQL). For example, I can get back data about a particular dish by sending a request to the URI that identifies it (e.g., http://api.menus.nypl.org/dishes/1860):
The highest 5-star rating would apply to data that includes links to other datasets. One objective of further curation work might be to discover and contribute links between the NYPL data and other open data sets to create something like the concordances that the Cooper Hewitt Labs have.
built for entities in their collections.

In many data curation scenarios the most urgent tasks involve moving data from the original site of creation to a stable environment (like a repository) where it can be preserved, but also where it can be reliably accessed. Neither of these problems is at issue with the menus data, so the potential curator can consider what other activities might improve the usefulness of the data.

**What’s a Data Curator To Do?**

Looking just beyond the (valid, important) tasks of preservation and basic access that are currently occupying many academic libraries entering the realm of data curation, interesting additional possibilities emerge for constructing what the work of data curation can be. I’m particularly interested right now in work that data curators can do to build secondary and tertiary resources—reference
materials, if you will—around data. I mean particularly reference materials that draw on the skills of people with training in library and information science, things like indexes. These types of organized systems of description can be one way to provide additional value over full text search (which, for many kinds of data sets, e.g., a table of numerical readings, is not particularly effective anyway).

How might this apply to the data from NYPL’s menu transcription site? For this exploratory data curation exercise, I’m setting myself the goal of seeing what can be done with the names of various dishes in *What’s on the Menu?* (surely, one of the main points of interest in this data set). The end product I’m imagining is a good index to the dishes represented in NYPL’s collection of menus. We could have an “authorized form” for each dish, keep track of any alternate forms, and begin to work out categories of related dishes.
From this, we could make some headway toward producing linked data from the menus data set—via concordances like the Cooper Hewitt’s—and we could also make our index of dishes available to others as a reconciliation service for cleaning and normalizing other data sets (using tools like Open Refine and the Open Knowledge Foundation Labs’ Nomenklatura).

NYPL has a data set that scores very well on an established scale of openness—the library provides access to machine-readable, structured data in a non-proprietary format—but further curatorial work can still improve the usefulness of this data by ordering and systematizing it at a layer beyond the technical structure of file format. The reason additional curation is needed has to do with the difference between strings and ‘things.’

‘Strings Versus Things’

Advocates of linked open data often use some
variation of the phrase ‘from strings to things’ in order to convey the basic motivation behind the technology. A Google search will turn up numerous examples. See, for example, this talk by Mia Ridge at a Linked Open Data in Libraries, Archives and Museums (LODLAM) workshop from last year (2012). As Ridge explains, > Computers think in strings (and numbers) where people think in ‘things’. If I say ‘Captain Cook’, we all know I’m talking about a person, and that it’s probably the same person as ‘James Cook’). The name may immediately evoke dates, concepts around voyages and sailing, exploration or exploitation, locations in both England and Australia… but a computer knows none of that context and by default can only search for the string of characters you’ve given it.

An inspection of the What’s on the menu? data set shows that we’re working with strings. A search for “Brussel sprouts” returns 611 results including at least 3 that look nearly identical but have different counts for the
numbers of menus on which they appear. For our reliable index of dishes we want to be working with things (where we can leverage those nice HTTP URIs to convey our specific meaning in machine-parseable terms). In the era of Google, this type of variation and duplication in search results is something to which researchers are accustomed and perhaps it even reintroduces a kind of serendipity, however, this feature of full-text search, which operates on strings, does make asking other questions of the data more difficult.

To appreciate the effect that going from strings to things can have, I extended my method of inspection from a single search to the whole data set. The front page of *What’s on the menu* proclaims that 1,260,150 dishes have been transcribed to date (this was in late July so slightly higher now). A quick look at the downloaded data, suggests that it might be more
accurate to say there are 1,260,150 instances of dishes in NYPL's system. There are only 469,357 entries (rows) in “Dish.csv” (again, for the July data)—each one representing a “dish” that has been given a unique identifier. (To quickly check myself, I looped through the CSV file and totaled up the values from the “times_appeared” column. The result—1,257,525 dish instances—is close enough to the published value to confirm my assumption.) So, really we have 1.2 million instances of 469K 394 thousand types of dishes. Given the example of the Brussel sprouts, I suspect that the number of types is actually lower still. 469K 394 thousand “dishes” is large enough to make for an interesting challenge but curation of this data to create a reliable index is only half as big a job as it appears from the web site.

Cracking open the data set and inspecting it is one way of assessing the need for curation and the likely amount of effort required—at
394 thousand data “points” or even 1.2 million
the data set is small enough to do this without stretching
common workflows or computational tools. (You can
open “Dish.csv” in Microsoft Excel, for example.) You could
make a similar determination about the curatorial actions
needed and the rough scale of the challenge without
opening any of the data files.

Part of the basic conceptual equipment of data curation
(as a meta-discipline) is a rough taxonomy of types of
data: observational, experimental, simulation, etc.
Data curation researchers have also developed some
cross-cutting ideas of “data levels”—from more “raw” to
more “cooked.” These terms come from a techno-scientific
context (data levels developed in the context of
work with earth-observing satellite imagery) but we can
also use them to reason about humanities data like
that from What’s on the menu by thinking about the project
like a system.

The NYPL has imaged a
collection of physical objects producing a first level of data (though not really “raw” in any deep sense, cf. “Raw Data” Is an Oxymoron). Then, through the construction of the What’s on the menu? site/application, the Library processed this first level of data with the aid of online volunteers (another term for “crowdsourcing” being “human computation”). What we can download as a data set is roughly this second level of data—transcriptions based on the images. Treating the contents of the downloaded CSV files as a kind of partially-processed observational data both helps us estimate error and variation (hello “human error”) and also think about how to plan for transformations and changes to the data set (will “authorized” forms fully supersede original forms?). Thus, we can reason from our theoretical knowledge of data and data curation to guide practical “hands on” action.

Next Steps
In the next post, I’ll describe some of the actual data munging work I’m doing to get closer to an index of dishes for *What’s on the menu?*. The data set of “dish” names is small enough to open in Excel but is big enough to challenge the normalizing functionalities of the more-powerful Open Refine. I found a way around this bottle neck and fell into a few other useful workflows along the way.

**UPDATE (2013-08-17):**
Corrected the number of rows in the downloaded CSV file.
Refining the Problem

More work with NYPL's open data, Part Two

In my last post, I described a speculative approach to the data generated by the New York Public Library’s *What’s on the menu?* project with the aim of identifying some data curation activities that might improve the usefulness of such an open data resource. After briefly assessing what I was seeing in the downloadable data set, I decided to work towards normalizing the names of the various “dishes” in the system. With this post, I want
to talk about how I’m accomplishing this using Open Refine. What I’m describing here is not the construction of a tool or even a particularly robust workflow but hopefully a deepening of the exploration.

**Strings are not what they appear**

The comma-separated value (CSV) file on “dishes” in the downloadable data set contains around 395,000 rows. Each row contains information about one “dish” that users of *What’s on the menu?* have transcribed from the digitized historical menus on the library’s site. For my immediate purposes here, what’s most significant is that the data appears to contain an identifier (e.g., 470291), from which an HTTP URI for the item can be straightforwardly derived (http://menus.nypl.org/dishes/470291), as well as a string purportedly representing the name of the dish.

I say “purportedly” because the only evidence we have that the data in the second
column is a name is the label of that column. (We’re also assuming that one row in the CSV file maps to one entity.) Assuming data to be exactly what it purports to be is a shaky proposition. The values in the “id” column check out as identifiers when we test them by constructing HTTP URIs and requesting the designated content. We can check the values in the “name” column by inspecting a few of them to see whether they conform to a colloquial understanding of names. We can also examine the transcription interface where these values were collected/created to try to assess whether anything there clearly maps to a concept of names. First, in addition to values like “Glace Framboise,” which seem relatively straightforward, we also have as values of “name,” strings like “ham steak, glazed pineapple rings, sweet potatoes, timbale of spinach a la financière.” The concept of “dish” is accommodatingly loose but there is some ambiguity whether this is one dish or two or three or four.
Second, we can see that the concept of “name” does not appear in the transcription interface. Volunteers are shown a section of the digitized image and given a text field with the instruction: “Write the dish here exactly as it appears. Don’t worry about accents.”


The instruction to “write the dish” sidesteps tricky decision-making that might lower participation—volunteers only need to reproduce the string of characters in the image exactly as it appears (this is not as simple as it might seem). The designers of What’s on the menu? take the results of this human computation and store it in a field labeled “name.” This undocumented conceptual leap from “transcribing” to
“naming” explains some of the quality issues with the data. Based on these two checks, we can’t assume that the values in the “name” column are all, in fact, names of dishes.

Of course, the New York Public Library (NYPL) Labs’ team is well aware that transcription interfaces need to be designed to scaffold complex tasks like the creation of high quality structured data. *What’s on the menu?* was the first of a string of successful projects. A more recent project using historical theater programs shows how NYPL Labs have designed a different interface to support volunteers in creating data about entities via transcription of images. The transcription interface of *Ensemble* asks volunteers to first decide “What type of info is this?” then displays relevant fields like “name” (explicitly labeled) based on the answer. In this later project, the data collection interface corresponds much more closely to the entity model represented in the underlying database fields.
The greater sophistication of the transcription platform in *Ensemble* helps to bridge the gap between transcribing from images and creating data about entities. NYPL has improved the algorithm at work in their human computation.

Screenshot of the transcription interface from *Ensemble*

Understanding how the data was created helps the prospective curator to realize that the data set from *What’s on the menu?* comprises not structured information about entities like “dishes” but rather partially-processed observations (transcriptions) of regions of high-resolution imagery. (Indeed the largest data file in the downloadable set is a mapping between “dish” identifiers and regions of images of particular menu
Taking the labels of fields and columns in “Dish.csv” as authoritative would assume a transformation of the data from one state to another (observation of image to property of entity) that has not actually occurred (or at least, has not occurred purposefully or uniformly throughout the data set). Getting to an authoritative index of “dish” entities will require further curation.

Space to improve

Knowing that we’re dealing with “observational data” in the form of transcriptions should incline us to treat the values of “name” skeptically. Also this knowledge does much to bolster the original assumption that the size of the curation challenge is not 1.2 million dishes to normalize as the What’s on the menu? site proclaims, or even 395,000-odd dishes as the number of rows in the CSV file would suggest, but some smaller proportion of even that smaller number.
Since the first step towards an index of dishes will involve cleaning up the variant strings found in the “name” column of “Dish.csv,” I turned to Open Refine, which proclaims itself “a free, open source, power tool for working with messy data,” as the most obvious fit for the job. A common workflow for using Open Refine (hereafter just “Refine”) to clean up messy data involves loading in data, optionally doing some global transformations, then using the tool’s powerful clustering functionality to group very similar values that may in fact be duplicates even if purely algorithmic processes can’t definitely identify them as such. To start, I’ll follow that general pattern.

Normalizing values using transformations will probably turn up some additional matches even prior to trying the clustering methods. These are basic things like trimming leading and trailing spaces and collapsing extraneous internal spaces (between words) but even small
whitespace variations prevent the computer from making exact matches between strings. Transformations are available in the dropdown menus in the header for each column.

I also transformed all values to lowercase to eliminate variations due to irregular or inconsistent capitalization. After just these procedures, the percentage of duplicate values for dish “name” rises from 0.009% (hardly worth expending effort to fix) to a little over 6% (perhaps small in itself, but the leap suggests that we’re a long way from the ceiling for improvement). In specific terms, this means that there are actually thirteen different identifiers assigned to the dish “cold roast beef” based only on
small variations in white space and capitalization not any genuine ontological difference.

**Powering the power tool**

At 395,000 rows, the *What’s on the menu?* data set is substantial but can still be opened in a common application like Microsoft Excel. However, a data set of this size, with so many potentially-unique values does pose challenges for Refine when trying to use more powerful functions like clustering. To get the benefit of Refine’s capabilities for cleaning up messy data, I needed to be able to interact with the application programmatically rather than only through the standard graphical user interface (GUI).

The first step in clustering involves generating what Refine calls “facets”—a list of the unique values for “name” found in the data set. Right away we run up against apparent limitations of the tool. After churning away for
a couple of seconds, Refine reports that there are “370004 choices total, too many to display” and offers a link where we could “Set choice count limit.” This is frustrating behavior—we can’t see even a partial list of facets and it’s not immediately obvious how to make forward progress from this point. If we click on the link to set a higher choice limit count—up to 371,000 from the default 2,000, to accommodate the variation in the menus data set—Refine will obligingly attempt to calculate and display this many facets. Most likely, the browser page will become unresponsive and raise an error. In modern browsers, processes in different tabs are isolated from each other so only the Refine tab should crash but, in older browsers, the whole application may crash at this point: *caveat emptor.*
The sidebar where we hoped to see a list of facets in

With only a few other

browser tabs open—rather

than the usual tens—this

process actually succeeds on

my machine after a minute or

so. Yet even if the faceting

process succeeds, the result

slows the interface to a

sluggish crawl and clustering

still crashes the whole

process. In a section titled,

“Miscellany” under the help

documentation for the

faceting function, the

maintainers indicate that you

can raise the choice count

limit but only “if you think

your computer can handle it.”

More interestingly, the help

documentation goes on to say, that “whether ‘your

computer’ can handle it or

not depends mostly on your
web browser.” I’m using Google Chrome on a fast machine so clearly, for the menus data, raising the choice limit count so far above the default is an unworkable hack. Nonetheless, this particular failure of the “power tool,” with the blame falling on the web browser, contains the seed of a “better” workaround.

This better workaround is predicated on understanding how the Refine application works. Refine uses a client/server application architecture but both the client (the web browser window as GUI) and the server live on the user’s computer. The server component is written in Java, while the front-end interface is written in Javascript. What’s failing when we try to cluster the values from the “name” column in the menus data is not the backend server application that is computing the facets but rather the Javascript application that manages displaying and updating the information. So, I reasoned I
might be able to work around the frustrations of the current GUI, if I could just interact programmatically with the server component. There are links to old discussion forum posts as well as a number of actual client libraries in various programming languages for interacting with the Refine server in the official documentation. Often the motivation for “scripting” an application like Refine is to enable it to be used in “batch” mode—that is, without human intervention. That’s not my motivation here. As the documentation points out, clustering, which relies on human value judgements for merging very similar data values, can’t be done in this kind of batch mode.

Since I was going to have to go to the trouble of this workaround for the standard Refine interface, I did ask myself whether I still needed Refine. The string manipulation above could have just as easily been accomplished using the basic libraries of almost any
programming language. A little digging around in the source code and some strategic googling revealed enough to quickly re-implement the main clustering method. So, I could have written more code and ended up with the same functionality I’ve gotten out of Refine so far. I suspect that down this path lies more and more re-inventing of the wheel. For this little speculative project, I am content to have both the standard browser-based interface as well as a programmatic interface open at the same time.

**Clusters on command**

I’ll discuss what this process says about the potential for curating the data from *What’s on the menu?* but I want to offer a few more details about working with Refine. This section may drift into technical detail but the client libraries that can be used to programmatically drive the Refine server are very poorly documented.

Based on what I read in the
discussion forums and on a “sniff test” of the few source code repositories, I settled on using one of the more recently updated forks of the Python Refine client library. These client libraries do not work with official APIs. As the creator of this Python library explains, these libraries reverse engineer the Refine application by snooping the traffic between the browser-based client and the local server using tools like HTTP Scoop. As I mentioned, documentation is very scant, and, at least in the case of the Python client, the code is not very idiomatic, which makes it a little more challenging to decipher. However, it works just fine for the purpose of patching around the problem of overloading the GUI with too many facets.

The Python client library assumes that Open Refine is installed and the server component is running at the default address (a different value can be passed in at initialization if necessary). Then, a couple lines are sufficient to setup a
connection to the Refine server.

```python
from google.refine import refine, facet

server = refine.RefineServer()
grefine = refine.Refine(server)
```

Most of the ‘commands’ return the raw JSON output that the server sends back. So, for instance, once I’m set up I can list the projects in my copy of refine—in this case just one comprising the August 1st version of the data set from *What’s on the menu?*

```json
{u'2310205155087': {u'created': u'2013-08-16T20:45:49Z',
    u'customMetadata': {},
    u'modified': u'2013-08-16T20:56:56Z',
    u'name': u'2013_08_01_07_05_00_data'}}
```

Figuring out how to do what I wanted (faceting and clustering on the values of “name”) took some experimentation—particularly in figuring out how to pick my way through the objects.
returned by some of the functions. For instance, to inspect the facets of the data, I had to get access to a dictionary called ‘choices’ that is part of the response to the ‘compute_facets’ function:

```python
facet_response = nypl_dishes.compute_facets(name_facet)
facets = facet_response.facets[0]

for k in sorted(facets.choices, key=lambda k: facets.choices[k].count, reverse=True):
    print(facets.choices[k].count, k)
```

This produces a list of unique values and their associated raw counts:

13 potatoes hashed in cream
13 cold roast beef
11 club sandwich
10 lobster salad
10 hot roast beef sandwich
10 american cheese
10 clams: little necks
9 celery
9 american cheese sandwich
9 strawberry ice cream

I created a quick iPython notebook to demonstrate the
details of how I’m using the client library to drive Refine.

**Return on investment**

The main thrust of this post has been demonstrating a “how to” for working around some of the limitations of Open Refine for cleaning and reconciling data like that from *What’s on the menu?*—where the size of the data set is in the hundreds of thousands of rows and where there is enough variation that the standard browser GUI cannot handle the load. The larger question is whether there is a still a plausible vision for how a data curator could add value to this data set. The need to script around limitations of a tool increases the cost of normalizing the NYPL data. At the same time, the ability to see the clusters of similar values that Refine produces increases my confidence that the potential gain in data quality could be very substantial in going from the raw crowdsourced data to an authoritative index. Using just the default method (usually
the most effective), produced 25 thousand clusters that need to be evaluated and reconciled. In future posts, I’ll report on how well the combination of the standard graphical interface and programmatic control are working to help me improve the NYPL’s data.
On Missing Data Sets

This repo will be periodically updated with more information, links, and topics.

Overview

What is a Missing Data Set?

"Missing data sets" are my term for the blank spots that exist in spaces that are otherwise data-saturated. My interest in them stems from the observation that within many spaces where large amounts of data are collected, there are often empty spaces where no data live. Unsurprisingly, this lack of data typically correlates with issues affecting those who are most vulnerable in that context.

The word "missing" is inherently normative. It implies both a lack and an ought: something does not exist, but it should. That which should be somewhere is not in its expected place; an established system is disrupted by distinct absence. Just because some type of data doesn't exist doesn't mean it's missing, and the idea of missing data sets is inextricably tied to a more expansive climate of inevitable and routine data collection.

Why Do They Matter?

That which we ignore reveals more than what we give our attention to. It’s in these things that we find cultural and colloquial hints of what is deemed important. Spots that we’ve left blank reveal our hidden social biases and indifferences.

Why Are They Missing?

There are a number of reasons why a data set that seems like it should exist might not, and they are all tied to the quiet complications inherent in data collection. Below are four reasons, with accompanying real-world examples.
1. Those who have the resources to collect data lack the incentive to (corollary: often those who have access to a dataset are the same ones who have the ability to remove, hide, or obscure it).

Police brutality towards civilians provides a powerful example. Though policing and crime are among the most data-driven areas of public policy, traditionally there has been little history of standardized and rigorous data collected about police brutality.

Nowadays we have a political and cultural climate where this issue has become one of public discussion. Public interest campaigns like Fatal Encounters and the Guardian’s The Counted have helped fill that void. But even for these individuals/organizations, the work is difficult and time-consuming. The group who would make the most sense to monitor this issue—the law enforcement agents who create the data set in the first place—have no incentive to actually gather such data, which could prove incriminating.

2. The data to be collected resist simple quantification (corollary: we prioritize collecting things that fit our modes of collection).

The defining tension of data collection is the struggle of taking a messy, organic world and defining it in formats that are neat, clean, and structured.

Some things are difficult to collect and quantify by nature of their structure. We don’t know how much US currency is outside of our borders. There’s no incentive for other countries to monitor US currency within their countries, and the very nature of cash and the anonymity it affords makes it difficult to track.

But then there are other subjects that resist quantification entirely. Things like emotions are hard to quantify (at this time, at least). Institutional racism is subtle and deniable; it reveals itself more in effects than acts. Not all things are easily quantifiable, and at times the very desire to render the world more abstract, trackable, and machine-readable is an idea that itself deserves questioning.

3. The act of collection involves more work than the benefit the presence of the data is perceived to give.

Sexual assault and harassment are woefully underreported. And while there are many reasons why this is, one major one is that in many cases the very act of reporting sexual assault is a very intensive, painful, and difficult process. For some, the benefit of reporting isn't perceived to be equal or greater than the cost of the process.

4. There are advantages to nonexistence.

Every missing dataset is a testament to this fact. Just as the presence of data benefits someone, so too does the absence. This is important to keep in mind.

However, there’s an even more specific angle to this point. To collect, record, and archive aspects of the world is an intentional act, one that typically benefits those who have the power to decide what should be collected. Often, remaining outside of the bounds of collection can be a form of response for a situationally-disadvantaged group. In short, sometimes a missing dataset can function as a form of protection.

Below is an ever-expanding list of missing datasets. A website with a full list is coming soon.

An Incomplete List of Missing Data Sets

This list will always be incomplete, and is designed to be illustrative rather than comprehensive. It also comes primarily from the perspective of the U.S, though the complete list of datasets features far more international examples.

- Civilians killed in encounters with police or law enforcement agencies [update: this is no longer a missing dataset]
- Sales and prices in the art world (and relationships between artists and gallerists)
• People excluded from public housing because of criminal records
• Trans people killed or injured in instances of hate crime (note: existing records are notably unreliable or incomplete)
• Poverty and employment statistics that include people who are behind bars
• Muslim mosques/communities surveilled by the FBI/CIA
• Mobility for older adults with physical disabilities or cognitive impairments
• LGBT older adults discriminated against in housing
• Undocumented immigrants currently incarcerated and/or underpaid
• Undocumented immigrants for whom prosecutorial discretion has been used to justify release or general punishment
• Measurements for global web users that take into account shared devices and VPNs
• Firm statistics on how often police arrest women for making false rape reports
• Master database that details if/which Americans are registered to vote in multiple states
• Total number of local and state police departments using stingray phone trackers (IMSI-catchers)
• How much Spotify pays each of its artists per play of song

Responses & Hypotheses

As part of my former Data & Society and current Eyebeam residency, I’m working on a number of projects that aim to consider possible responses to missing data sets. This will be updated with documentation of those projects as they are published.

• **Data won’t solve all problems.** Data are useful for informing a debate, increasing knowledge, shaping a conversation, and providing context. Data can give the ability to have knowledge about trends, and how things have changed over time. But having data isn’t enough to solve all problems (just because we now know how many people are killed in moments of police brutality doesn’t mean that police brutality has ended.)

• **Collective action is a strategy for resistance.** My hypothesis is that one answer to these missing datasets lies in those who have a stake in the data cooperating to disrupt the structures preventing access to it, a la Jonah Bossewitch and Aram Sinnreich’s sousveillance society model (see Resources folder for paper).

• **Lack of collection is also a strategy.** This has been said before, but bears repeating. A tricky aspect of dealing with missing data sets is that they hint at larger problems, and the answer to those problems does not universally lie in collecting more data.

Resources

**Links**

"The Point of Collection" - a piece I wrote for Data & Society’s Points publication that expresses much of the conceptual background for this.

"The Detroit Geographic Expedition and Institute: A Case Study in Civic Mapping" - Catherine D'Ignazio's excellent case study on the Detroit Geographic Society, which was a fascinating 1960s response to missing data.

"Where We Live and How We Die" - article for How We Get To Next that explores death and data, highlighting how difficult it is to talk about both as a result of missing data.
You Say Data, I Say System

In the fall of 2009, I wrote a pair of algorithms to place nearly 3,000 names on the 9/11 memorial in Manhattan. The crux of the problem was to design a layout for the names that allowed for what the memorial designers called ‘meaningful adjacencies’. These were requests made by next-of-kin for their family members to appear on the memorial next to—or as close as possible to—other victims. Siblings, mothers and daughters, business partners, co-workers, these connections represented deep affinities in the real world. There were nearly 1,400 of these adjacencies that a layout of the names would ideally honour.

In December of that year, I flew to New York to meet with some of the project’s stakeholders and to present the results of the algorithms that I’d developed. I came into the meeting disheveled and nervous. Disheveled because I’d flown into La Guardia that morning, having spent much of the plane ride revising and re-revising my presentation. Nervous because I had found out the day before that another team had
also been working on the layout problem; a group of financial analysts ('quants') who almost certainly all had at least one PhD.

It must’ve been a strange sight. A small army of besuited financial professionals, across the table from a long-haired artist from Canada with an old, broken laptop. The quants went first: they’d run permutation after permutation on their server clusters, and they were confident they’d found the optimal solution for the adjacencies: a maximum about 93 percent of them could be satisfied. They’d asked to speak first because they wanted to ‘save us all some time’, since they knew, mathematically, that they had found the most highly optimized solution.

It was a persuasive argument. I let them finish, then I turned my laptop around on the table to show them a layout that I’d generated about a week before—one that was 99.99% solved.

The lesson here is not ‘don’t get a math PhD’. Nor is it (specifically) ‘hire a long-haired data artist from Canada’. The lesson is to not look just at the data, but at the entire system that the data is a part of. Taking a systems approach to data thinking allows you not only to solve problems more efficiently, but to more deeply understand (and critique) the data machinery that ubiquitously affects our day-to-day lives.
An over-simplified and dangerously reductive diagram of a data system might look like this:

\[ \text{Collection} \rightarrow \text{Computation} \rightarrow \text{Representation} \]

Whenever you look at data—as a spreadsheet or database view or a visualization, you are looking at an artifact of such a system. What this diagram doesn’t capture is the immense branching of choice that happens at each step along the way. As you make each decision—to omit a row of data, or to implement a particular database structure or to use a specific colour palette you are treading down a path through this wild, tall grass of possibility. It will be tempting to look back and see your trail as the only one that you could have taken, but in reality a slightly divergent you who’d made slightly divergent choices might have ended up somewhere altogether different. To think in data systems is to consider all three of these stages at once, but for now let’s look at them one at a time.

**Collection**

Any path through a data system starts with collection. As data are artifacts of measurement, they are beholden to the processes by which we measure. This means that by the time you look at your .CSV or your .JSON feed or your Excel graph, it has already been molded by the methodologies, constraints, and omissions of the act of collection and recording.

The most obvious thing that can go wrong at the start of a data system is error, which is rife in data collection. Consider the medical field: A 2012 study of a set of prestigious East Coast hospitals found that only 3% of clocks in hospital devices were set correctly, meaning that any data carrying a timestamp was fundamentally incorrect. In 2013, researchers in India analyzed results from the humbly analogue blood pressure cuff in hospitals and clinics and found the devices carried calibration errors in the range of 10% across the board.

These kinds of measurement errors are pervasive, inside of hospitals and out. Errors may be unintended, the results of mis-calibrated sensors, poorly worded surveys, or uncounted ballots. They can also be deliberate, stemming from purposeful omissions or applications of heavy-handed filters or conveniently beneficial calibrations.
Going further back from *how* the data is collected, you should also ask *why*—or *why not*. Artist and data researcher Mimi Ohuoha, whose practice focuses on missing data, tells us that the very decision of what to collect or what not to collect is political. “For every dataset where there’s an impetus for someone not to collect”, she writes, “there’s a group of people who would benefit from its presence”. Onuoha neatly distilled the importance of understanding collection to the understanding of a data system as a whole in her recent talk at the Eyeo Festival in Minneapolis: “If you haven’t considered the collection process”, she stated neatly, “you haven’t considered the data.”

**Computation**

After collection, data is almost certainly bound to be computed upon. It may be rounded up or down, truncated, filtered, scaled or edited. Very often it’ll be fed into some kind of algorithmic machinery, meant to classify it into meaningful categories, to detect a pattern, or to predict what future data points from the same system might look like. We’ve seen over the last few years that these algorithms can carry tremendous bias and wield alarming amounts of power. But this isn’t another essay about algorithmic bias. There are many other aspects of computation that should considered when taking the measure of a data system.

In Jacob Harris’s 2015 essay *Consider the Boolean*, he writes about how seemingly inconsequential coding decisions can have extraordinarily impact on the stories our data might ultimately tell. Harris proposes that the harsh true-false logic of computation and the ‘ideal views’ of data that we endeavour to create with code are often insufficient to represent the ‘murky reality the data is trying to describe’. Importantly, he underlines the fact that while computational bias can come from big decisions, it can also come from small ones. While we urgently need to be critical of the way we our author machine learning systems, we also need to pay attention to the impact of procedural minutiae—like wether we’re storing a data point as a boolean or a string.

**Representation**

As you’ve seen, the processes of collection and computation are rampant with decision points, each of which can greatly increase or greatly limit the ways in which our data systems function. When we reach the representation stage, and begin to decide how our data might
tell its story to humans, possibility space goes critical. Each time you pick a chart type or colour palette or a line weight or an axis label, you're trimming the possibility space of communication. Even before that, the choice of a medium for representation has already had a predestinatory effect. A web page, a gatefold print, a bronze parapet—each of these media is embedded with its own special opportunities, and its own unavoidable constraints.

Whatever the medium, many of the points that Mimi Onuoha makes about collection can be mapped directly to visualization: questions about what is shown in a visualization and how it is shown must be paired with questions about what isn’t shown and why someone has chosen not to show it. In a quest to avoid the daunting spectre of bias, data visualization practitioners often style themselves as apolitical. However, the very process of visualization is necessarily a political one; as I’ve said for years to my students at NYU, the true medium of data visualization is not color or shape; it’s the decision.

By being mindful of the decisions we’re making when we’re authoring visualizations we can make better work; by seeing these decisions in work made by others we can be more usefully critical of the data media that we consume.
The problem with the way that most of us operate within a data system is that we’ve designed our roles such that we can almost never see the whole thing from where we are. Those who are tasked with collecting the data are rarely involved in its representation. Conversely, the visualization professional sits so far away from measurement that most times the nuances of how the data was collected are completely lost. No matter where you might reside on the collect/compute/represent continuum, it will do you service to stand on high ground, to stretch your vision as far as you can towards the opposite edge.

Which brings us back to the memorial, to the algorithm, to all of those names.

Where the quants failed, I think, was in not considering the physicality of the memorial itself. They looked at the data, but not at how the data was to be represented. While their model generalized the problem, considering each name as an equal unit in a simulated system, mine considered each name as a unique unit in a real system. My model rendered each name using the typeface in which the name would ultimately be inscribed in bronze. It considered the half-inch expansion joints between the memorial’s parapets, and how the individual characters in each specific name might allow it to cross that expansion joint neatly. It included the oddly-shaped triangular corners of the memorial along with the long rectangular parapets. While each of these strange characteristics of the physical memorial might seem like a constraint, they also gave my system an elasticity that couldn’t exist in a simplified model.

Since my work on the memorial, I’ve tried to cultivate a data systems approach in all of the work that I’ve done with my many collaborators. I’ve gone deep into systems of collection by building sensor stations on glaciers, narrowly escaping hippo attacks and riding in submersibles to the bottom of the ocean. I’ve built web tools and machine learning systems to compute upon billions of records of web advertising placements to provide means for collective action against discrimination. I’ve explored the outer limits of data representation through sound, sculpture, performance, and participatory practice.

But embracing a data systems approach doesn’t need to be so involved; it can be a simple act of word replacement. The next time you read a story with the word data in the headline, swap it out with data system.
When you see a data visualization, think of it instead as a data system visualization. If the government proposes new policies around personal data, think about them instead as policies about people, and the data systems which they inhabit. Widening your thinking in this fashion will also allow you to engage in broader criticism of data systems and those who are authoring and exercising them.

After all, it's not enough for us to be critical of Uber's booking algorithm, or FOX News' most recent infographic. We need to expand our attention to the systems that these mechanisms support; systems in which our participation is often both transparent and involuntary. By taking a systems approach to data I believe we can make better things. And we might also find deeper and more meaningful questions—questions that are as much about how these things work (or don't work) as why they exist in the first place.

---

The images in this post are by Harry Fisk, and were made in 1944. They show the historic meanderings of the Mississippi river. They are the result of years of research, and are, as Kyle Hill writes “a combination of speculation, interpretation, and extrapolation”. Find out more about them here: http://nerdist.com/harold-fisks-incredible-maps-track-the-ghosts-of-the-mississippi/
Practitioners, critics, and popularizers of new methods of data-driven research treat the concept of “data cleaning” as integral to such work without remarking on the oddly domestic image the term makes—as though a corn straw broom were to be incorporated, Rube-Goldberg-like, into the design of the Large Hadron Collider. In reality, “data cleaning” is a consequential step in the research process that we often make opaque by the way we talk about it. The phrase “data cleaning” is a stand in for longer and more precise descriptions of what people are doing in the initial phases of data-intensive research. If you work with data or pay attention to discussions among practitioners who do, you’ve probably heard or read somewhere that 80 percent of that
work is “cleaning”.\(^1\) Subsequently you likely realize that, there is not one single understanding of what “data cleaning” means. Many times the specifics of “data cleaning” are not described anywhere but reside in the general professional practices, materials, personal histories, and tools of the researchers. That we employ obscuring language like “data cleaning” should be a strong invitation to scrutinize, perhaps reimagine, and almost certainly rename this part of our practice.

The persistence of an element that is “out of focus” in discussions of data-intensive research does not invalidate the findings of such research, nor is it meant to cast researchers using these methods under suspicion. Rather, the collective acceptance of a connotative term, “cleaning,” suggests two assumptions: first, that researchers in many domains consider the consequences of whatever is done during this little-discussed 80 percent of the process devoted to “cleaning” as sufficiently limited or bounded so as not to threaten the ultimate value of any findings; and second, relatedly, that there is little to be gained from more precise description of those elements of the research process that currently fall under the rubric of “cleaning.”
As researchers working in the relatively new domain of data-intensive research in the humanities, we have found that these assumptions do not serve us well. In fields where data intensive work has a longer history, researchers have developed paradigms and practices that *de facto* define “data cleaning.” However, in the humanities, these bounds are still unformed. Yet the humanities cannot import paradigms and practices whole from other fields, whether from “technoscience” or the nearer “social” sciences, without risking the foreclosure of specific and valuable humanistic modes of producing knowledge. If we are interested in working with data and we accept that there is something in our work with data that is like what other fields might call “data cleaning,” we have no choice but to try to articulate both what it is and what it means in terms of how humanists make knowledge.

This may be a only a current issue, a tax on those humanities researchers who wish to adopt new methods, asking them to over-explain their work processes in order to hash out new regimes for research in this domain. Once new methods are more widely practiced, the data-intensive humanities researcher may also be able to toss off
the shorthand of “data cleaning.” For now, there is value in being arrested by the obfuscation of this phrase. Trying to more precisely say what we mean by “data cleaning” can be fruitful because this effort directs our attention to an unresolved conversation about data and reductiveness. In turn, this might help us to develop new work that blends the tradition of cultural criticism from the humanities with research that is also digital and data-intensive.

**Humanities Data and Suspicions of Reduction**

When humanities scholars recoil at data-driven research, they are often responding to the reductiveness inherent in this form of scholarship. This reductiveness can feel intellectually impoverishing to scholars who have spent their careers working through particular kinds of historical and cultural complexity. The modern humanities have invested mental and moral energy into, and reaped insights from, studying difference. Bethany Nowviskie summarizes this tradition in her contribution to the (forthcoming) 2017 edition of *Debates in the Digital Humanities*, “Capacity Through Care.” Nowviskie writes:
The finest contribution of the past several decades of humanities research has been to broaden, contextualize, and challenge canonical collections and privileged views. Scholars do this by elevating instances of neglected or alternate lived experience—singular human conditions, often revealed to reflect the mainstream.

From within this worldview, data cleaning is then maligned because it is understood as a step that inscribes a normative order by wiping away what is different. The term “cleaning” implies that a data set is “messy.” “Messy” suggests an underlying order. It supposes things already have a rightful place, but they’re not in it—like socks on the bedroom floor rather than in the wardrobe or the laundry hamper.

Understood this way, suspicions about “cleaning” are suspicions that researchers are not recognizing or reckoning with the framing orders to which they are subscribing as they make and manipulate their data. In fields where researchers have long been explicit about their framing orders, the limits of results are often
understood and articulated using specialized discourses. For example, in climate science, researchers confine their claims to the data they can work with and report results with margins of error. While humanities researchers do have discourses for limiting claims (acknowledging to the choice of an archive or a particular intellectual tradition), the move into data intensive research asks humanists to modify such discourses or develop new ones suitable for these projects. The ways in which humanities engage these challenges may both open up new practices for other fields and allow humanities researchers who have made powerful critiques of the existing systems of data analysis to undertake data-intensive forms of research in ways that don’t require them to abandon their commitments to such critiques.

To contribute to the development of new discourses and the practice of critically-attuned data work, we scrutinize “cleaning” through a reflection on our own work with Curating Menus. Curating Menus is a research project that aims to curate and analyze the open data from New York Public Library’s What’s on the Menu.
The Value of a Naive Tool

We set off to answer questions like: can we see the effect of wartime food rationing in what appeared on menus during World War I? or, can we track the changing boundaries of what constituted a “dish” over time? To do this, we thought we needed to “clean” the “messy” data. What became evident was that “cleaning up” or “correcting” values was a misleading—and even unproductive—way to think about how to make the data more useful for our own questions and for other scholars studying food.

Under the rubric of “cleaning,” we began with a technical solution to what we’d imagined was a technical problem. Variation in the strings of text transcribed from menus was obscuring our ability to do things as simple as count how many dishes were in the data set. Trevor, along with Lydia Zvygintseva, began trying to reduce the number of variations using Open Refine. When the scale of the problem overwhelmed the capabilities of that tool, Trevor discovered that it was possible to run the clustering algorithms popularized by Open Refine using custom Python scripts. The
output of one of these scripts were lists of variant values such as:

<table>
<thead>
<tr>
<th>id</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2759</td>
<td>Potatoes, au gratin</td>
</tr>
<tr>
<td>7176</td>
<td>Potatoes au Gratin</td>
</tr>
<tr>
<td>8373</td>
<td>Potatoes--Au gratin</td>
</tr>
<tr>
<td>35728</td>
<td>Potatoes: au gratin</td>
</tr>
<tr>
<td>44271</td>
<td>Au Gratin Potatoes</td>
</tr>
<tr>
<td>84510</td>
<td>Au Gratin (Potatoes)</td>
</tr>
<tr>
<td>94968</td>
<td>Potatoes, au Gratin,</td>
</tr>
<tr>
<td>97166</td>
<td>POTATOES:- Au gratin</td>
</tr>
<tr>
<td>185040</td>
<td>Au Gratin [potatoes]</td>
</tr>
<tr>
<td>313168</td>
<td>Au Gratin Potatoes</td>
</tr>
<tr>
<td>315697</td>
<td>(Potatoes) Au Gratin</td>
</tr>
<tr>
<td>325940</td>
<td>Au Gratin Potatoes</td>
</tr>
<tr>
<td>330420</td>
<td>au-Gratin Potatoes</td>
</tr>
<tr>
<td>353435</td>
<td>Potatoes: Au gratin</td>
</tr>
<tr>
<td>373639</td>
<td>Potatoes Au Gratin</td>
</tr>
</tbody>
</table>

We were very excited to get lists that looked this way because we could easily imagine looping over such lists and establishing one normalized value for each set of variants. We hadn’t yet recognized that the data model around which the data set was organized was not the data model we needed to answer our research questions. The main challenge seemed to be processing enough values quickly enough to “get on with it.”
At this point, the Python scripts we were using were small, purpose-built command line programs. After some deliberation, we decided to build a simple web application to provide the task-specific user interfaces we needed to tackle the challenge of NYPL’s menu data.³

The piece of software we built does, in some ways, the opposite of what one might expect. A cluster of values like the one for “Potatoes Au Gratin” above is presented to the user, and he or she (Trevor or Katie) have to make a decision about how to turn that cluster
of variants into a single value. Our tool sorts the variants by the number of times they appear across the data set. So the decision may be to simply save the most commonly occurring value as the normalized form: “potatoes au gratin”. Or it might be to modify that value based on a set of rules we have created (more on that later). Or it might be to supply a new value. The process can end up looking like this:

... What would be the authoritative spelling of Buzzards Bay oysters? Let me Google that.

... Oh, it collapsed an orange juice and an orange sherbet under “orange”; let me flag that.

... A jelly omelet!?

The tool surfaces near-matches, but it does not automate the work of collapsing them into normalized values. Instead, it focuses one’s attention and labor on exactly that activity. In our initial version of computer-assisted data curation, you still have to touch each data point.

In the process of normalizing values, we found ourselves faced with questions about the foods themselves. Choosing
the “correct” string was not a self-contained problem, but an issue that required returning to our research questions and investigating the foods themselves. Since the research questions we were asking required us to maintain information about brands and places, we often had to look up items to see which was the correct spelling of the brand or place name. Shellfish and liquor were two particularly interesting and plagued areas for these questions. The process revealed kinds of “messiness” that we had not yet even considered. We realized the data points we were making were not “corrections” “cleaning up” the original data set, but their rather formed an additional contribution of information with its own data model. What we were developing was not an updated version of the NYPL’s data set. What we thought were data points were, in fact, a synthesis or mash-up of different kinds of information within a half-finished or half-articulated data model. We needed to create our own data set, which would work in context with the NYPL data set.

This approach is made possible by and explicitly rests on a structure of linked data. The NYPL data was created to be
linkable—it uses unique URLs for dishes and menus. Our data can include links referencing these URLs. The linked data paradigm thus encourages us to build our own data and to (inter-)link it with the original.

**Diversity in Data**

The Curating Menus data set is an organized hierarchy of concepts from the domain of food. To make it, we attach labels that we believe will best facilitate researchers’ understanding of the scope, diversity, and value of the NYPL’s data set for research. Our interaction with the NYPL data set became a process of evaluating variants. Which variants in the names of dishes revealed new information we should account for in our own data, and which variants were simply accidents of transcription or typesetting? The process freed us to attend to difference and detail rather than only attempting to hide it or clean it away. We can be sensitive to questions about time, place, and subject. This kind of attention is imperative if humanities researchers are to find the menus data legible.

As we considered methods for preserving diversity within our large
data set, the work of anthropologist Anna Tsing offered a valuable theoretical framework to approach these issues. In “On Nonscalability: The Living World is Not Amenable to Precision-Nested Scales,” Tsing critiques scalability as an overarching paradigm for organizing systems (whether world trade, scientific research, or colonial economies). By scalability, Tsing means the quality that allows things to be traded out for each other in a totalizing system without regard to the unique or individual qualities of those things—like many stalks of sugarcane (which are biological clones of one another), or, subsequently, workers in a factory. From this definition of scalability, she goes on to argue for a theory of nonscalability. Tsing writes, “The definition of nonscalability is in the negative: scalability is a distinctive design feature; nonscalability refers to everything that is without that feature … Nonscalability theory is an analytic apparatus that helps us notice nonscalable phenomena.” While scalable design creates only one relationship between elements of a system (what Tsing calls “precision nesting”), nonscalable phenomena are enmeshed in multiple relationships, outside or in tension with the nesting frame. “Scales jostle and contest each
other. Because relationships are encounters across difference, they have a quality of indeterminacy. Relationships are transformative, and one is not sure of the outcome. Thus diversity-in-the-making is always part of the mix.”

Currently, the imagination of the cultural heritage world has been captured by crowdsourced information production on the one hand and large-scale institutional aggregation on the other—the What’s On the Menu? project exemplifies both of these trends. Our difficulties working with the open data from this project suggest that it is a vital moment to consider the virtues of non-scalability theory in relation to digital scholarship. Engineering crowdsourced cultural heritage projects usually involves making object transcription, identification, and the development of metadata scalable. For example, the makers of the What’s On the Menu? project designed their system to divide the work into parcels that could be done quickly by users while reducing friction that arise from differences in the menus (the organization of the information on the page, other evidence of physical manifestations like handwriting and typeface variations).
The images of menus and the metadata about them are also being republished through projects like the Digital Public Library of America (DPLA), another example of how things get shaped and parsed for purposes of scaling up to ever wider distribution. Tsing reminds us, “At best, scalable projects are articulations between scalable and nonscalable elements, in which nonscalable effects can be hidden.” She argues that the question is not whether we do or do not engage in scalable or non-scalable processes. To explore the articulations between scalable and nonscalable, Tsing tells the story of the contemporary matsutake industry, which encompasses both foraging (by immigrant harvesters in the ruins of large-scale industrial forestry in the U.S. Pacific Northwest) and global supply chains serving Japanese markets. Tsing’s account focuses our attention on how “scales... arise from the relationships that inform particular projects, scenes, or events” (509). The elements of nonscalable systems enter into “transformative relationships” and these “contact[s] across difference can produce new agendas” (510). Following Tsing, we came to see points of articulation which had previously been invisible to us as would-be consumers of scaled data. Beginning from the creation of the original, physical menus
and tracing the development of the crowd-created data, we identify and account for “nonscalable elements”—and consequently, edge further and further from the terminology of “cleaning.”

**Seeing Nonscalability in NYPL’s Crowdsourced Menus Project**

Making menus is a scalable process. Although menus are sometimes handwritten or elaborately printed on ribbon-sewn silk, the format of a menu is designed to be scalable. Menus are an efficient typographical vehicle for communicating a set of offerings for often low-margin dining enterprises. Part of the way that we know that menus are scaleable is how alike they appear. “Eggs Benedict” or “caviar”, with their accompanying prices may fit interchangeably into the “slots” of the menu’s layout. Within the menus themselves, we also see evidence of the nexus of printing scalability, dish scalability, and cost in, for example, the use of ellipses to express different options: eggs with ... cheese, ... ham, ... tomatoes, etc. The visual evidence of *What’s on the Menu?* shows us how headings, cover images, epigraphs—for all their surface variations—follow recognizable patterns. These strong
genre conventions and the mass production of the menus as physical objects allow us to see and treat them as scaled and scalable.\(^7\)

Scalability in the original physical menus.

However, the menus also express nonscalable elements—historical contingencies and encounters across difference. Some of these nonscalable elements are revealed by the kind of questions we find ourselves asking about the experience of ordering from these menus. How were they understood as part of interactions between purveyors, customers, and diners? How did diners navigate elements like the pervasive use of French in the late nineteenth and early twentieth centuries? How did they interpret the particular style and content of cover images or quotations? Evidence for these questions manifests in the menus as objects but does not fit within the scalable frames of menu production nor the menu data we have at hand. The nonscalable elements cannot be disregarded and have the potential to impact how we interpret
and handle the scalable data. Nonscalability theory encourages us to grapple with this dynamic at each point of articulation in the process of making scalable objects.

The collection of these menus was also scalable. The system set up for their accession and processing not only treated the menus as interchangeable objects; it also treated them like the many other paper materials that entered the collections of the New York Public Library in the twentieth century. Perhaps the clearest evidence of this is in the cataloging. The catalog cards fit each menu into the same frame—with fields for the dining establishment, the date of creation and date of accession, and the sponsor of the meal if available. Cataloging is a way of suppressing or ignoring the great differences in the menus, choosing one type of data to attend to. The cards index the collection so that the institution has a record of its holdings and so that a user can find a particular object. The menus, with their scalable and nonscalable features, become scalable library inventory through this process.⁸

Cataloging’s aim is to find a way to make items at least interchangeable enough not to break the system. The
practice is rife with examples of catalogers navigating encounters with difference. Catalogers practice nonscalability theory constantly. Sometimes the answer is institutionally-specific fields in MARC records; sometimes the solution is overhauling subject headings or creating a new way of making records (like the BIBFRAME initiative). However, the answer is almost never to treat each object as a unique form; instead the object is to find a way to keep the important and usable information while continuing to use socially and technologically embedded forms of classifying and finding materials.

Digitization is also a process designed for scalability. As long as an object can fit into the focal area of an imaging device, its size, texture, and other material features are reduced to a digital image file. The zooming enabled by high-resolution digital images is one of Tsing’s prime examples of design and engineering for scalability. In the distribution of digitized images, the properties of the digital surrogate which are suited to scalability are perpetuated, while the properties of the original which are nonscalable (the feel of the paper, its heft or daintiness) are lost.9
The point at which certain objects are selected for digitization is one of the moments of articulation Tsing describes between the scalable and nonscalable. Digitization transforms of diverse physical materials—brittle, acidic paper or animal parchment, large wooden covers or handstitched bindings, leaves or inserts—into standardized grids of pixels. From the point of digitization forward, the logic of scalability permeates projects like *What’s On the Menu?*. The transcription platform is constructed to nest precisely within the framework of how cultural heritage organizations like NYPL create digital objects from their original materials.

Paul Beaudoin from the NYPL Labs team discusses some of the logic behind their approach to these kind of projects in a blog post announcing Scribe, an open-source software platform released in 2015 but derived from the library’s experience with crowdsourced transcription projects. Beaudoin describes how the Scribe platform is based on “simplification [that] allows us to reduce complex document transcription to a series of smaller decisions that can be tackled individually. ... the atomicity of tasks makes projects less daunting for
volunteers to begin and easier to continue.” What’s on the Menu?, for example, presents visitors with a segment of a digitized image of a menu and a single text input box to record what they see. The NYPL Labs team is explicit about its commitments to designing for scalability. We know from work in the domain of scholarly editing that what comprises “transcription” is not self-evident. It could be modeled and implemented in software in a number of ways. The menus project uses Optical Character Recognition (OCR) software to generate bounding boxes that determine what human volunteers will transcribe. In this, we can see the “precision nesting” of scales at work. OCR software is designed to scalably produce machine-readable, machine-processable digital text from images of printed material. In the case of the menus, the software can detect regions of text within the digital images; however, due to the variation in typefaces, the ageing of inks and paper, and other nonscalable elements, the output of the OCR algorithm is not a legible set of words. Using the bounding boxes but discarding the OCR text in favor of text supplied by human volunteers is a clever and elegant design. It constructs the act of transcription in such a way that it matches the scalable process of
digitization and ways of understanding the content of a menu that privilege scalable data.

Yet, even here, now that we know to look for them, the nonscalable effects cannot be completely hidden. The controls allow users to zoom through three levels of an image, a feature that evidences slippage in the segmentation algorithm. This element of the tool acknowledges that someone might need to zoom out to complete a transcription—often because the name of a dish has to be understood through the relation between the line of text in the bounding box and other nearby text, like a heading. Further, the text box on the transcription screen is unadorned, implying that what to do is self-evident, but the help page is full of lengthy instructions for how to “navigate some of the more commonly encountered issues,” revealing the ways that transcription is not a self-evident, scalable process.
The physical layout of text often requires zooming out the image presented for transcription to decipher a dish.

In addition, the project was designed so that people did not have to create accounts or sign in to submit transcriptions. This reflects a view of volunteers as interchangeable, and embedded in this assumption is the hope that it allows more work to get done more quickly. However, what is construed as a scalable workforce is, in fact, made up of people who have different levels of understanding or adherence to the guidelines and different perceptions or interpretations of the materials. When we understand this workforce as a collection of individuals, we can see how any crowd as large as the one that has worked on the menus project will contain such diversity. The analytic apparatus of Tsing’s nonscalability theory makes all these design choices visible and allows us to see the transcription task, as framed within What’s on the Menu?, as another moment of articulation between scalable and nonscalable elements.

When we download the open data from the What’s on the Menu? site and open up the files, we are presented
with the results of all this activity—menu collection and digitization and transcription. Instead of seeing mess, we see the ways in which diversity has seeped or broken into what were designed to be smoothly scaling systems. Now we are better prepared to envision how our work—creating new data organized around concepts of historical food practices—begins from what the NYPL has released, which is transcription data (words volunteers typed into the boxes at What’s on the Menu? linked to metadata from their digital library systems). In both of these data sets there is something called “dish.” In NYPL’s data, “dish” is the name of the field in which a transcribed string from a menu is stored in the project’s database. In Curating Menus’s data, “dish” is a representation created to reflect and name an arrangement of foods and culinary practices in a particular historical moment. This is an example of, as Tsing puts it, the ways that “scales jostle and contest.” We know that the response to this friction is not to retreat from working at scale. Instead we have to find ways of working while aware that precision nesting hides diversity and that there are stakes to things being hidden.
Indexes: Making Scalability Explicit and Preserving Diversity

Our answer this challenge is an index. We’re suggesting that indexing is a more precise replacement for some of the work that travels under the name of “cleaning.” An index is an information structure designed to serve as a system of pointers between two bodies of information, one of which is organized to provide access to concepts in the other. The lists of terms and associated page numbers from the back of a book is one familiar example. An array of other terms that people use alongside “cleaning” (wrangling, munging, normalizing, casting) name other important parts of working with data, but indexing best captures the crucial interplay of scalability and diversity that we are trying to trace in this piece.
We began to think of the work we were doing as building something like a back-of-the-book index for the *What’s On the Menu?* data. We would create additional data structures around and atop the existing data, generating links between a set of names or categories we created and the larger and more heterogeneous set of data from NYPL. We are interested in ultimately building two interconnected indexes, one focused on historical food concepts and one on the organizations connected to the menus (businesses, community organizations, fraternal societies, etc.). We have begun with the food index, and we are developing a framework that echoes cookbook indexes in order to structure our data: ingredients, cooking methods, courses, cuisines.

If we felt no unease continuing the lineage of precision nesting that link the scales of digitization and crowd-sourced transcription, we could proceed with a completely algorithmic approach—“cleaning” our data using scripts, linguistic rules, and even machine learning. These methods yield results by providing an approximation—which we suspected might hide diversity. We could imagine trusting that an approximation could be good
enough or using algorithmic approaches. However, to understand what was being approximated—and what was being smoothed together—we needed to create a grounded approach to making a data model for our index.

Now when we look at those lists of variations on “Potatoes au Gratin” or some other group of transcriptions, we are focused on the task of choosing a label that will be a node in our data set and will serve as a pointer to all of the varying transcribed values in the NYPL data set. We are building the set of labels from the set of data rather than beginning by writing some large, hierarchical domain model. We want to represent the concept of two eggs and bacon not caring if it was written “bacon and two eggs” or “2 Eggs and Bacon.”

To get from transcription to concept, we began with a set of simple rules: spell out numbers, use lower case letters. Actually engaging with the menu transcriptions quickly raised other questions. For example, on the question of place names, we decided to apply capitalization rules (in accord with style guides like the *Chicago Manual of Style*) that say that you
capitalize when the reference to place is literal, but not when the reference makes a generic association: yes to Virginia ham or Blue Point oysters but no to swiss cheese or scotch. We also found many single transcriptions containing multiple concepts, like “steak, sauce béarnaise.” Since we want a way to be able to systematically find multiple components of a dish, we’re opting to standardize how we labeled the addition of sauces, garnishes, and other added ingredients. Here is one instance where we plan to use algorithmic tools to help us analyze some of this big data after we have grounded it in a specific data model.

In building an index, we are engaged in creating scalability. We know that scalability is a process of articulations between different scales; however, Tsing suggests—and we believe—that those articulations are often hidden. Conversely, indexes are tools of scalability that make these articulations explicit.

Our index is about ingredients, meal structures, and cooking techniques. Someone else could re-index the menus material in a different way. Variations might involve attending to the species of the plants and animals
that are in foods or taking a nutritional approach that classifies food based on calories, vitamins, carbohydrates. We can also imagine projects that attend to language use in ways that our index suppresses. As libraries and researchers move forward in making and curating data, instead of the constant refrain of “cleaning,” we want to encourage indexing, which allows us to build up explicit and flexible bases of knowledge that people can continue to access and understand.

**Sharing Control of Authority**

One of the mechanisms that librarians and archivists have used to build and maintain large, distributed information systems is a set of practices referred to as authority control. In brief, these practices involve creating defined and agreed upon taxonomies as well as guidelines for the application of such arrangements of terms. The Library of Congress Subject Headings represent one instance of authority control. Maintaining such a system is labor intensive and has been used only for supporting core library activities like managing collections and supporting patrons in finding materials. Libraries and archives are trying to take advantage of technological
developments in linked data—merging their centuries-old authority control practices with the affordances of the World Wide Web. However, what relatively few have seized on are new opportunities to apply the practices of authority control outside the original core needs of collection organization and wayfinding.

These new opportunities fall somewhere between digital library practices and digital humanities research, but the gap is one that more projects should embrace the opportunity to fill. There is a need for projects that take “authority work” as an invitation to new creativity; an invitation for making and building. In such a model, multiple regimes of authorities might be built up from critically-aware and engaged intellectual communities to meet their own specific needs while also participating in larger technological and information systems.

We imagine those communities will contain librarians and scholars. Though librarians and humanities scholars have frequently intersected, they have rarely interacted in the ways we are calling for. Simplifying to the point of caricature, these existing interactions go something like this: humanities
scholars point out that the structure and content of a specific archive or collection represents even recreates certain cultural logics. For example, the systems of describing collections—such as the widely-used Library of Congress Subject Headings—reify concepts about persons or cultures that really ought to be interrogated more closely or perhaps discredited and dismantled all together. For the most part, librarians, archivists, and information scientists acknowledge these flaws and perhaps even work to remedy them in the course of maintaining systems that preserve whatever partial archives do exist or helping patrons to find information they need.

We are looking for new forms of collective action that can be expressed through the professional work of humanities scholars and librarians. This is not simply a call for the production of more and more data—attempting to subvert the work of categorization and classification through the production of ever more local, finely-wrought distinctions, details, qualifications. Our aim is to develop ways of working that validate local experiences of data without removing them from a more global network of information.
exchange. These practices, as we imagine them, resonate with Bethany Nowviskie’s interpretation of the Afrofuturist philosophy of Sun Ra (as expressed by Shabaka Hutchings), which claims: “Communities that have agency are able to form their own philosophical structures.” The transition to working in a linked data paradigm should be valued not principally for the ways in which it might make large-scale information systems operate more smoothly, but rather for the ways in which it can create localized communities of authority, within which people can take control of the construction of data and the contexts in which it lives. In a keynote presentation at the 2015 LITA Forum Mx (Mark) A. Matienzo articulated a parallel version of this view, saying:

We need to begin having some serious conversations about how we can best serve our communities not only as repositories of authoritative knowledge or mere individuals who work within them. We should be examining the way in which we can best serve our communities to support their need to tell stories, to heal, and to
work in the process of naming.

Discussions of “cleaning” data fails to capture this need. The cleaning paradigm assumes an underlying, “correct” order. However tidy values may look grouped into rows or columns or neatly-delimited records, this tidiness privileges the structure of a container rather than the data inside it. This is the same diversity-hiding trick that nonscalability theory encourages us to recognize.

It is not enough to recognize; we also wish to offer a way of working. In arguing against cleaning, we propose index-making. In this approach, the first things we would do with our data sets, rather than normalize them, is find the communities within which our data matters. With those communities in mind and even in dialogue, we would ask, what are the concepts that structure this data? And how can this data, structured in this way, point to other people’s data? This way of thinking allows us to see the messiness of data not as a block to scalability but as a vital feature of the world which our data represents and from which it emerges.
Please cite as


Notes


2. The fact that these communities have developed discourses for describing the boundaries of the claims they make does not inoculate them from critique about the costs and shortcomings of their methods. Cf. Bruno, Latour, “Why has critique run out of steam? From matters of fact to matters of concern.” Critical Inquiry 30.2 (2004): 225-248.

3. For a variety of reasons, we would not recommend this course of action to others without serious deliberation. There is a reason why applications like Open Refine are so popular and useful. If you would like to know more, contact us.

4. If we had a dictionary to compare these materials too, the process may have been
more automatable; however, from what we have found thus far, that particular language resource—Wordnet for Oysters!—doesn't exist.↩


7. The NYPL menu collector Frank E. Buttolph’s acquisition practices reinforce the role and scale of printers in menu production in the twentieth century. In addition to restaurants and customers, she went straight to the menu source—printers—to fill out her collection.↩

8. Cf. Tsing 519↩


10. Early versions of the project interface featured a social-media-style call-to-action below the image snippet (“What does this
say?”), as well as brief instructions below the text input box: “Please type the text of the indicated dish EXACTLY as it appears. Don’t worry about accents” (see an example at the Internet Archive). This accompanying text was quickly dropped—presumably because the task seemed self-evident enough from the layout of the transcription screen.

11. See https://datasymposium.wordpress.com/sahle/.
Presentation given on Saturday, April 27, 2013, at HASTAC 2013 in Toronto, Canada.

You will notice I have changed the title of my presentation a bit from what is in the program. Partially, it’s because I’m an indecisive academic, but mostly, it is in reaction to my experience of co-hosting a HASTAC forum on “Visualization Across Disciplines” this past week. There has been some amazingly rich conversation thus far it’s still open for participation. So, instead of talking about my own research (which is on the simultaneous analytic, aesthetic, and social use of data visualization), I’m going be a bit more theoretical and hopefully a bit more thought provoking.

As a new media scholar with one foot in visualization and the other in the digital humanities, I often find myself asking myself this:

“What exactly is visualization in the digital humanities?”

We’ve already established and can agree upon why we use it.

But what I’m more interested in and what we never really talk about is the how. How do we use visualization in the digital humanities? How does it function at the level of epistemology? Is it a tool? A
research lens? A communication medium? Something else? What I’m going to do in this presentation is focus on the first option -- a tool -- and try to expand the way we think about visualization in the digital humanities to something beyond this construct.

First, let me clarify what I mean by visualization. Here are two definitions common to the visualization field as it has developed over the last 20 or so years.

We could certainly come up with others; but what I want to note are two important points: 1) the visualizations I’m talking about are digital, and 2) they help us to “make sense” of data.

Usually, we think of visualization in the digital humanities, we think of it as a digital tool. This comes as no surprise given the field’s historical origins. Early digital humanities projects, like John Burrow’s textual-analysis of 17th and 18th century verse used visualization complemented by statistics to help “make sense” of the large volumes of data now available to humanities inquiry.

The culture of digitization that characterized the digital humanities through the 2000s only magnified these volumes, and scholars increasingly began use visualization to look not only at textual content but also spatiotemporal data, non-text artifacts and related metadata over the longue duree.

Over the last 10 or so years, countless digital humanities projects have pushed visualization’s humanistic application in this functional tool-driven way. A good example is “Mapping the Republic of Letters.” The project, which was not incidentally developed as part of Stanford’s Tooling Up for Digital Histories Project, uses visualization to help scholars explore the over 55,000 letters and documents digitally archived in the Electronic Enlightenment Database. By “mapping out” geographic and related data for senders and receivers of letters from the early modern period, it allows researchers to perceive larger patterns of intellectual exchange.

“Mapping the Republic of Letters” was created for the particular project from the ground up, but we could just as easily look any of the many projects created by scholars using one publicly available visualization tool.
toolkits – such as Gephi, NodeXL, ManyEyes, Google Fusion Tables, the list goes on.

Or perhaps the ultimate example of visualization’s role as “tool” within the field is its integration within software and research environments designed specifically for humanistic research. TAPoR 2.0, RoSE, Lev Manovich’s ImagePlot (a tool that I helped develop) or Tanya Clement’s still under development platform for humanities visualization.

In each of these examples, what visualization is allowing us to do is to:

- extend our conceptual scope and reach
- create and discover new knowledge
- and then also represent this process

But digital tools are often more than tools (this is one of the big ideas of HASTAC, right)? It’s not just that visualization is a graphical or cognitive aid to thinking. It is thought itself.

This is not a new idea. It’s an extension of what was originally proposed by Rudolf Arnheim in his 1969 book Visual Thinking. In it, Arnheim argues that all perceiving is thinking and all thinking is also perceiving. The two, as he puts it are “indivisibly intertwined.”

The idea has since been expanded by many scholars in the name of information visualization, media studies, contemporary technogenesis, theories of extended cognition… and in our forum this past week under the guise of a conversation about process.

The basic idea is this. We think of data visualization as a tool that gives us a product (a.k.a. insight). But it’s not just in the perception of this product that we gain insight, it’s also in the process of its creation. As Elijah Meeks has commented, “Dataviz isn't just a product, but oftentimes it's the exposed computational process.” This is where the visual thinking occurs.
Unfortunately, there’s often a gap between perceiver and creator, between representation and process that makes this visual thought appear unbiased, intuitive, and largely positivistic – all characteristics that stand in marked contrast with the type of uncertain and interpreted data we encounter in the humanities.

So the question becomes, how to we build this iterative process (including the data wrangling, the active visual thinking, and the experience of coming to knowledge) into the representation of visualization?

The most interesting example that came up in conversation (thanks to Mia Ridge) was a Lattice Uncertainty Visualization created by researchers at the University of Calgary and University of Toronto. It’s essentially a visualization that sits on top of a translation algorithm for an instant message conversation between a German speaker and an English speaker. What happens is that the German speaker types a message in German and what comes out on the other screen is this. Every path through the lattice represents a hypothesis about the translation. The varying transparency of each node reveals the certainty of each word (with the dark blue being more certain). The user can then go through and interactively change the horizontal green path through the lattice to indicate a better translation.

What I find fascinating about this visualization is not only that it reveals the uncertainty of the algorithm and the influence of viewer interpretation. It essentially forces the viewer to go through the creation and reception of the visualization process (even if it is just only a small part of the process).

Opinions? Reactions? Ideas about how to incorporate process into visualization? And so I’ll end with the question I began with…What is visualization in the digital humanities? I hope that I’ve made this a little more difficult to answer.
Visualization as a Digital Humanities ________ ? | HASTAC

Tags

process
HASTAC Conferences & Events
HASTAC Scholars

Groups

Scholars
Scholar Class 2013

HASTAC Conferences

- HASTAC 2017: The Possible Worlds of Digital Humanities
- HASTAC 2016: Impact, Variation, Innovation, Action
- HASTAC 2015: The Arts and Science of Digital Humanities
- HASTAC 2014: Hemispheric Pathways: Critical Makers in International Networks
- HASTAC 2013: The Storm of Progress: New Horizons, New Narratives, New Codes
  - Keynotes
    - Conference Schedule
    - Videos
  - Related Programming
  - Post-Conference Workshops
- Reflections on HASTAC 2013
  "I no longer feel that we are so very few": HASTAC and Community Building
  A #HASTAC2013 Interactive Archive [Updated]
  A networked HASTAC 2013
  HASTAC 2013 - Innovation as Activism: Ideas for Libraries
  HASTAC 2013 post-report
  HASTAC Poetry Contest Winners
  HASTAC2013 conference: seeking shelter - mobilizing creative and archival practices
  Navigating Encounters w/ Neatline: Geotemporal History and the Battle of St. George's Key

https://www.hastac.org/blogs/tzepel/2013/05/02/visualization-digital-humanities
Notes Towards a Regional Digital Arts & Cultures Collaboratory; or, HASTACing With the Neighbors
Reflection on the Storm of Progress
Roundup of HASTAC 2013 posts
Unlearning Teaching
Visualization as a Digital Humanities ?

Abstract: Alternative Sprawls, Junkfutures: Buenos Aires Libre’s Urban Epistemology
The Calm Before the Storm: Reflecting on HASTAC 2013 "The Storm of Progress"
Conference Before the Start of the Fall Semester
Voice without a Budget: Digital Technologies in a Community College Classroom
What HASTAC Has Taught Me About Love in the Academy
Where is the Story?
CFP: HASTAC Conference 2013

Digital Scholarly Communication: Conference Proceedings from HASTAC 2011
HASTAC 2010: Grand Challenges and Global Innovations
HASTAC 2009 Conference Buzz
HASTAC II: Techno-Travels (2008)
Information Year (2007)
Conference Planner To-Do List

Similar Content

"Searching" Victorian Print: The Process of Joining Text and Illustration
Visualizing Schneemann
On Mostly Cartographic Education
Patrik Svensson on the Role of Public Education and the Future of Digital Scholarship
Spotlight series #2: Mobile histories
WHAT WOULD FEMINIST DATA VISUALIZATION LOOK LIKE?  
Submitted by kanarinka on December 20, 2015 - 8:33am

Seeing the whole world is a fantasy that Michel DeCerteau calls the "totalizing eye" and Donna Haraway calls "the God Trick". This is the first image taken of the whole earth in 1967. From [Wikipedia](https://en.wikipedia.org/wiki/First_image_of_the_earth_taken_from_space).

In January, I'm headed to the [Responsible Data Forum's event about Data Visualization](https://responsible-data.org/). While there is a lot of hype about data visualization, and a lot of new tools for doing it (my colleague Rahul Bhargava and I have counted over 500!), fewer people are thinking critically about the politics and ethics of representation. This, combined with a chart-scared general public, means that data visualizations wield a tremendous amount of rhetorical power. Even when we rationally know that data visualizations do not represent "the whole world", we forget that fact and accept charts as facts because they are generalized, scientific and seem to present an expert, neutral point of view.

What's the issue? Feminist standpoint theory would say that the issue is that all knowledge is socially situated and that the perspectives of oppressed groups including women, minorities and others are systematically excluded from "general" knowledge. Critical cartography would say that maps are sites of power and produce worlds that are intimately bound up with that power. As Denis Wood and John

---

CATHERINE D'IGNAZIO
Research Affiliate

Catherine D'Ignazio is the person behind that really cute baby. She is an Assistant Professor of Data Visualization and Civic Media at Emerson College who investigates how data visualization, technology and new forms of storytelling can be used for civic engagement.

Professor D'Ignazio has conducted research on geographic bias in the news media, developed custom software to geolocate news articles and designed an application, "Terra Incognita", to promote global news discovery. She is working on sensor journalism around water quality with PublicLab, data literacy projects and various community-educational partnerships with her journalism students. Notably, she co-organized a hackathon at the MIT Media Lab called "The Make the Breast Pump Not Suck!" Hackathon.

Her art and design projects have won awards from the Tanne Foundation, Turbulence.org, the LEF Foundation, and Dream It, Code It, Win It. In 2009, she was a finalist for the Foster Prize at the ICA Boston. Her work has been exhibited at the Eyebeam Center for Art & Technology, Museo d’Antiochia of Medellin, and the Venice Biennial.

Professor D'Ignazio is a Fellow at the Emerson Engagement Lab and a
Krygier note, the choice of what to put on a map "... surfaces the problem of knowledge in an inescapable fashion as do symbolization, generalization and classification". Until we acknowledge and recognize that power of inclusion and exclusion, and develop some visual language for it, we must acknowledge data visualization as one more powerful and flawed tool of oppression.

Can we say this more vividly? Donna Haraway, in her seminal essay on Situated Knowledges, offers a brilliant tour-de-force critiquing not just visual representation but the extreme and perverse privileging of the eyes over the body that has dominated Western thought. If you could, dear reader, read this quote aloud as it truly functions as a piece of performance art:

The eyes have been used to signify a perverse capacity - honed to perfection in the history of science tied to militarism, capitalism, colonialism, and male supremacy - to distance the knowing subject from everybody and everything in the interests of unfettered power. The instruments of visualization in multinationalist, postmodernist culture have compounded these meanings of disembodiment.

The visualizing technologies are without apparent limit. The eye of any ordinary primate like us can be endlessly enhanced by sonography systems, magnetic resonance imaging, artificial intelligence-linked graphic manipulation systems, scanning electron microscopes, computed tomography scanners, color-enhancement techniques, satellite surveillance systems, home and office video display terminals, cameras for every purpose from filming the mucous membrane lining the gut cavity of a marine worm living in the vent gases on a fault between continental plates to mapping a planetary hemisphere elsewhere in the solar system.

Vision in this technological feast becomes unregulated gluttony; all seems not just mythically about the god trick of seeing everything from nowhere, but to have put the myth into ordinary practice. And like the god trick, this eye fucks the world to make techno-monsters.

-- Donna Haraway in "Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspective/Feminist Studies" (1988)

The God Trick! Is this not the rhetorical premise and the seductive promise of most data visualization? To see from the perspective of no person, no body? Our appetite for such perspectives is fierce, "gluttonous", as Haraway characterizes it.

And yet, there are ways to do more responsible representation. There are ways to "situate" data visualization and locate it in concrete bodies and geographies. Critical cartographers, counter-mapping artists, indigenous mappers and others have experimented for years with these methods and we can learn from them.

Here are some beginning design thoughts about what feminist data visualization could do:

1. Invent new ways to represent uncertainty, outsides, missing data, and flawed methods

While visualizations - particularly popular, public ones - are great at presenting wholly contained worlds, they are not so good at visually representing their limitations. Where are the places that the visualization does not go and cannot go? Can we put those in? How do we represent the data that is missing?

Andy Kirk has an incredible talk about the Design of Nothing that surveys the field in regards to how designers make decisions about representing uncertainty, including zeros, nulls and blanks. Can we push more designers to take these methods into consideration? Can we ask of our data that it point to its own outsides?
Map to Not Indicate, 1967, by the art collective Art & Language. The map depicts only Iowa and Kentucky and then proceeds to list the many things that NOT represented on it. Part of the Tate Collection.

Beyond simply missing data - how do we dig into data provenance as an entire subfield of visualization akin to the reporter's work of fact-checking and verification? Can we collect and represent the data that was never collected? Can we find the population that was excluded? Can we locate the faulty instrument that everyone assumed was working? Can we critically examine the methods of a study rather than accepting the JSON, CSV or API as is? This may seem like it's not the designer's job. Someone else prior to them in the pipeline will do that un-sexy investigative work of data anthropology. But if data visualizers don't take on this responsibility, who does?

2. Invent new ways to reference the material economy behind the data.
Akin to this question of data provenance, we also need to ask about the material economy behind the data. What are the conditions that make a data visualization possible? Who are the funders? Who collected the data? Whose labor happened behind the scenes and under what conditions?

For example, the Public Laboratory for Open Technology and Science (a civic science group with whom I am an organizer) has a technique of mapping where you hang a camera from a kite or balloon in order to collect aerial imagery. One side effect of this method that the community has embraced is that the camera also captures imagery of the people mapping. These are the bodies of the data collectors, so often absent from final representations.
From a Public Lab research note by Eymund Diegal about mapping sewage flows in the Gowanus Canal. Note the people on boats doing the mapping and the balloon tether that links the camera and image back to their bodies.

Data visualizations often cite data sources as fact on a legend but we could do more. What if we visually problematized the provenance of the data? The interests behind the data? The stakeholders in the data? A single CSV file or streaming feed often has no reference to any of these more human, material elements that are nevertheless essential to understanding the where, why and how of data.

Perhaps one way to solve this would be to have, by default, or to collect oneself, much more robust metadata and to intentionally prioritize the visual display of that metadata. The goal of such visualizations would be to show not just "what the data says" but to show how the data connects to real bodies, systems and structures of power in the wider world.

3. Make dissent possible
While there are plenty of "interactive" data visualizations what we currently mean by this is limited to selecting some filters, sliding some sliders, and viewing how the picture shifts and changes from one stable image to another stable image as a result. These can be powerful methods for diving into a contained world that consists of stable images and stable facts. But as we know from Wikipedia editing wars and GoogleMap Controversies the world is not actually bracketed so conveniently and "facts" are not always what they appear to be.

So one way to re-situate data visualization is to actually destabilize it by making dissent possible. How can we devise ways to talk back to the data? To question the facts? To present alternative views and realities? To contest and undermine even the basic tenets of the data's existence and collection? A visualization is often delivered from on high. An expert designer or team with specialized knowledge finds some data, does some wizardry and presents their artifact to the world with some highly prescribed ways to view it. Can we imagine an alternate way to include more voices in the conversation? Could we effect visualization collectively, inclusively, with dissent and contestation, at scale?

What else?
These are just three design suggestions that point towards a feminist ethics and politics of data visualization. What else? I'd love to hear what other aspects of data visualization we could re-think to make it more situated, more feminist and ultimately, more responsible. Post your thoughts here in the comments or @kanarinka on Twitter and let's start the conversation.

References:


Optional Readings
Blunt Instrumentalism: On Tools and Methods

DENNIS TENEN

I am on the side of the makers. I believe that the humanities can be a place not just to think about things, but to do things. Doing, when done right, can expand the scope of our critical activity, prepare our students for work in the world, and finally—and this despite the protestations of some—enact meaningful change in our communities (Fish). I write, then, being inspired by research at institutions such as the Critical Making Lab at University of Toronto, Concept Lab at UC Irvine, and metaLab at Harvard, along with many similar research centers that routinely engage with material culture as a matter of scholarly practice. In my courses as well, students create models, curate exhibitions, file patents, convene conferences, write grant applications, send letters to the Senate, draw, build, and code. However, the academy also presents some unique challenges to critical making of that sort, particularly when it comes to sustainable tool development. As tool makers, we should heed the lessons of the numerous forgotten projects that did not find an audience or failed to make an impact. For every line of code actively running Pandoc, NLTK, or Zotero, there are hundreds that lie fallow in disuse. Yet even in failure, this codebase can teach us something about the relationship between tools and methods.¹

In reflecting on my own failed projects, I have come to believe that with some notable exceptions, the university is an unfit place to develop "big" software. We are much better poised to remain agile, to tinker, and to experiment. The digital humanities (DH) can be understood as part of a wider "computational turn" affecting all major disciplines: see computational biology, computational linguistics, computational social science, computational chemistry, and so on. Computation in the humanities supplements the traditional research toolkit of a historian, a literary scholar, and a philosopher.² In this chapter, however, I would like to bring into question a specific mode of tool making, practiced within the digital humanities and without, of the kind that confuses tools with methods. The tools I have in mind prevent or—more perniciously—tacitly discourage critical engagement with methodology. To discern the problem with tools more clearly, imagine a group of astronomers using a telescope that reveals to them wondrous star constellations. Yet our hypothetical scientists cannot tell if these stars actually exist or whether they are merely an artifact of a faulty telescope. This has always been the tool-wielder’s dilemma. Contemporary research instrumentation in our field, from natural language processing to network analysis, involves complex mechanisms. Their inner workings often lie beyond the full comprehension of the casual user. To use such tools well, we must, in some real sense, understand them better than the tool makers. At the very least, we should know them well enough to comprehend their biases and limitations.
The best kind of tools are therefore the ones that we make ourselves. After spending days wrangling a particularly messy corpus, I might write a script that automates data cleanup. My code may strip out extraneous HTML markup, for example. I could then release the script as a software library to help others who face the same task. With time, I might add a graphical user interface (GUI) or even build a website that makes using my scripts that much easier. Such small acts accelerate the research capabilities of the field as a whole. I would do nothing to discourage analogously altruistic sharing. But let us be sure that in using tools we also do not forget to master them from the inside out. What if my code implicitly mangles important metadata; or worse, what if it alters primary sources in an unexpected and tendentious ways? Let the tool makers make such biases explicit to the public.

Methods Within

Some tools encourage intellectual laziness by obscuring methodology. More often, it is not the tool but rather a mode of lazy thinking that is at fault. For example: the `nltk.cluster` module bundled in Python’s Natural Language Toolkit (NLTK) framework (Bird, Klein, and Loper) contains an implementation of something called “k-means clustering,” an unsupervised method of finding groups of similar documents within a large collection. The “unsupervised” part means that we are looking for hidden structure without making any assumptions about the documents at the outset (Na, Xumin, and Yohng). The documents may be grouped by the preponderance of personal pronouns or perhaps by sentence length. We do not know what elements the algorithm will identify, only that it will make piles “typical” of our corpus. The tricky part comes in estimating the number of expected document clusters (that is the $k$ variable). In a corpus of nineteenth-century novels, for example, one may expect a dozen or so clusters, which could perhaps correspond to novelistic genres. When clustering a large database of diplomatic communiques, one would reasonably expect more fine-grained “piles” of documents, which could have something to do with regional differences or with major political events. In either case, the algorithm will blindly return some groupings of distinctly related documents. But whatever the results of clustering, they are difficult to interpret in terms of meaningful literary-historical categories like “genre” or “period.” Some of our piles will correspond to genres and periods, while others will seem meaningless. The algorithm produces nonhierarchical results—that is, the output is not ordered according to value or significance. As the algorithm is also nondeterministic, meaning that it will perform differently each time it is run, the groupings will also vary with each iteration. To complicate matters, NLTK implements other clustering algorithms, like expectation–maximization (E-M) and group average agglomerative clustering (GAAC). These methods will likely chance upon yet other hidden relations between documents and other ways of organizing the material into piles. The algorithm will always return a result, according to some set of formal commonalities. But what these results mean and why they matter is open to interpretation. To make the clusters
NLTK facilitates such discovery by distributing detailed documentation along with the code. The documentation does more than just describe the code: it reveals implicit assumptions, citing external sources throughout. In experimenting with NLTK, I was able to get some output from the clustering methods in a matter of days. It took me months to understand what they could mean and how they could be applicable to my research. Just applying the tool or even “learning to code” alone was therefore insufficient for making sense of the results. What could help me, then, and what is only now beginning to surface in DH literature is a critical conversation about methodology.

Unlike some other tools of its kind, NLTK is particularly good at revealing its methods. Its codebase is open to inspection; it is easy to read; and it contains much commentary along with links to related research. The NLTK project began in 2001, at the University of Pennsylvania, in a collaboration between a linguist and his student (Loper and Bird). Research based on the module started appearing in print several years later, around 2004. NLTK reached version 1.0 eight years after its inception, in 2009. In the intervening time, immense care must have went into the critical apparatus that ships with the tool. And I suspect that at this late stage of the project, more hours have gone into the writing of its documentation than into the crafting of its code. As of 2015, the NLTK GitHub page lists no fewer than 130 contributors.

Reflecting on the history of NLTK gives us a glimpse into the realities of responsible academic making. Not every project will need to go through such a long development cycle or include such detailed documentation. But even my own small collection of data cleaning scripts would need substantial work to reach the level of polish required for empowered use of the kind NLTK enables. Note also that NLTK itself is only a “wrapper” around a set of statistical methods for the analysis of natural language. That layer of encapsulation already poses a number of problems for the researcher. Using NLTK responsibly demands a degree of statistical literacy along with programming experience. The cited methodology often contains a mixture of code and mathematical formula. Yet higher-level encapsulations of NLTK, like a web-based topic modeler, for example, would further remove the user from that implicit logic. Each level of abstraction in the movement from statistical methods, to Python code, to graphical user interface introduces its own set of assumptions, compromises, and complications. Any “ease of use” gained in simplifying the instrument comes at the expense of added and hidden complexity.
Hidden complexity puts the wielder of the tool in danger of resembling a hapless astronomer. To avoid receiving wondrous pictures from broken telescopes, in the way of actual astronomers, we must learn to disassemble our instruments and to gain access to their innermost meaning-making apparatus. Any attempt to further repackage or to simplify the tool can only add another layer of obfuscation.

It follows, then, that without a critical discussion about implicit methods, out-of-the-box tool use is best treated with a measure of suspicion. The makers of out-of-the-box tools should similarly weigh the altruistic desire to make research easier against the potential side effects that come with increased complexity. The tool can only serve as a vehicle for methodology. Insight resides in the logic within. When exposed, methodology becomes subject to debate and improvement. Tools proliferate and decline in quality relative to the researcher's experience. If tomorrow's scholars move from Python to Haskell, the effort of learning the underlying algorithms is what will transfer with the language. Methodology is what remains long after the tools pass into obsolescence.
In addition to methodological concerns, tool making also involves pragmatic considerations about sustainability. Software is cheap and fun to build by contrast to the expense and drudgery of its maintenance. "Ninety percent of coding is debugging. The other 10 percent is writing bugs." The aphorism comes naturally to program managers and software engineers who have gone through the full software product development cycle. In the excitement of building new tools, it is however easy to underestimate the challenges of long-term application maintenance. Academic attention spans are naturally cyclical: articles are published, interest wanes, funding dries up, students graduate. Scholars start anew each year and each semester. Software support requires the continuity of care and much more of it as a codebase matures. Standards change, dependencies break, platforms decay, users have questions. The case for the humanities as a laboratory for innovation is strong, but I doubt that many are prepared to make "critical customer support" a part of their research agenda.

Software development requires immense resources, as digital humanists from George Mason and the University of Virginia will tell you. Smaller teams should think twice before investing time and money into tool development. Not every method needs to be packaged into a tool. Some projects would be better off contributing to existing efforts or using their resources to encourage methodological literacy. In fact, if you build it, they might not come at all. Start-ups know that beyond the initial excitement of a product launch, the challenge of any new application lies in the acquisition and the retention of users, no matter how "disruptive" or "innovative" the technology.

A few years ago, I spent some time working with a talented French developer to design a collaborative translation platform. Despite his skills and dedication to the project, the tool did not gain significant traction among language teachers, translators, or students. I learned then that no amount of innovative engineering or beautiful web design could guarantee participation. Neither of us had the time nor the resources to advocate for the service. Advocacy would require arranging for training, outreach, fundraising, and support: services we could not provide in addition to our professional obligations. It was however tempting to think that social and institutional change could ride on the coat tails of software alone. If we build it right, the two of us thought, we could transform the practice of translation in the classroom. Yet we failed to consider the difficulty of implementing that vision into practice. We built the tool but not the community around it. The classroom environment resisted change, and for a good reason. Upon reflection, we saw that language teaching was grounded in proven, if sometimes imperfect, practices. Our platform development should have considered the strengths of that tradition and not just its weaknesses. Before rushing to innovate, we could have started with smaller classroom experiments to test our intuitions. We could have arranged for interviews, focus groups, and pilot studies. To give you a sense of our miscalculation, consider Duolingo, a similar (and earlier) effort led by researchers from Carnegie Mellon University, which amassed more than four million dollars of investment...
In retrospect, it was hubris to attempt platform building without similar commitments.

Consider also the following in the case of our hypothetical “wrapper” around NLTK—the one that would simplify the use of natural language processing for the nontechnical audience. Every contemporary Mac and Linux laptop machine comes prepackaged with powerful command-line tools for text manipulation: software utilities like `wc`, `sort`, and `uniq`. When chained together, these simple instruments are used to count and sort words in a document or to generate a term-frequency distribution useful for formal text analysis. They are free, simple to learn, versatile, and require no additional installation. They come with their own textbook, accessible from the terminal. Yet most of my students, even at the intermediate level, remain unaware of such tools already at hand. Many were not exposed to the basics of file paths, networking, or operating systems. How can one better facilitate the practice of computational text analysis without closing the digital literacy gap that separates mere users from empowered tinkerers and tool makers? A proposal to implement yet another tool duplicating the functionality of ubiquitous native utilities gives me pause. We must first reflect on the reasons as to why there was no adoption in the first place. That is not to say that existing word-frequency tools cannot be refined in some way. But, any new project that hopes to innovate would have to at least match the power of the existing instrumentation and then improve on it in some palpable way. And even then, our hypothetical project would face the same barriers to literacy and adoption as the original toolkit. These would have to be addressed before writing a single line of code.

Furthermore, whatever adoption the new alternative might achieve risks fracturing the existing user base, already limited to a small number of practitioners. By analogy, a new publishing platform that hopes to uniformly “disrupt” academic publishing is far more likely to enter an already fragmented market rife with good alternatives that are struggling to survive. The fragmentation prevents any one them from gaining critical mass. Instrumental efficacy alone therefore cannot address the lack of adoption. For example, legacy platforms like Microsoft Word or clunky journal management systems (used behind the scenes for peer review) do not account for the range of “planned obsolescence” problems in academic publishing that Kathleen Fitzpatrick identified in her recent book on the subject. The tool comprises but a small part of a much larger publishing ecosystem. It can act as a wedge that initiates change, but not without a larger communal effort to address the way we read, write, and do research. The world does not suffer from a lack of better text editors, for example. Rather, the adoption of powerful free and open source software is stymied by insufficient training, institutional momentum, and the lack of intellectual buy-in. Rather than fracturing the community, by creating another text editor for example, we would often do better to join forces: to congeal our efforts around common standards and best practices. Unfortunately for us, funding agencies favor promises of bold innovation where it would be more prudent to
invest into organic growth. The effort to shift the habitus of a community, as Pierre Bourdieu would describe it, involves a delicate balance between disruption and continuance. Much can be learned from the success of the open-source and free culture movements in this regard (Weber). Take, for example, the story of Wikipedia and MediaWiki. MediaWiki, the software platform powering Wikipedia, was neither the first nor the most technically sophisticated wiki software package. But in the hands of Wikipedians, MediaWiki became a tool capable of transforming the contemporary information landscape. Despite some of its problems, Wikipedia struck the right balance between traditional forms of knowledge-making such as the encyclopedia and innovative editorial structures such as commons-based peer production. Wikipedia the community inspires me more than MediaWiki the tool. In the Wikipedia world, the platform is secondary to community development.

The care of academic research communities, of the kind that encourages empowered tool use, happens in departments and through professional organizations. Programs like the Digital Humanities Summer Institute answer the need for training necessary to do research in our rapidly developing field. However, more resources are needed to initiate methodological and not just instrumental innovation. Few humanities-based alternatives exist to institutional structures in other fields like the Society for Political Methodology and the International Association of Legal Methodology; journals like Sociological Methods & Research, Journal of Mixed Methods Research, and International Journal of Qualitative Methods; prizes and funding opportunities like the Political Methodology Career Achievement and Emerging Scholars Awards, or the Program for Promoting Methodological Innovation in Humanities and Social Sciences administered by the Japan Society for the Promotion of Science. To sharpen our tools we must similarly prioritize methodological development. Only then can we build platforms that answer to the values of humanistic critical inquiry.

A shared concern with data and computation has brought a number of disciplines closer together. Biologists, linguists, economists, and sociologists increasingly integrate their methodologies, as evidenced by a vigorous cross-disciplinary publishing record. DH is primed to join that conversation, but only if its methods develop without abridgment. Tools are great when they save time, but not when they shield us from the complexity of thought. Working as a digital humanist or a new media scholar means taking on extra responsibilities: to do well by history when writing history, to do good science when doing science, and to engineer things that last when making things.

Notes

1. William Pannapacker has written eloquently on the topic in the Chronicle of Higher Education. See “Pannapacker from MLA: ‘The Success of Failure’.”

2. I do not mean to imply that DH can be reduced to computation. See Ramsay and Rockwell, “Developing Things,” and also Elliott, MacDougall, and Turkel, “New Old Things.”
4. The quote is commonly attributed to Bram Cohen, the creator of BitTorrent, posted on Twitter in 2011. There are however numerous earlier instances of the exact quote, itself a variation of Sturgeon's Law coined by Theodore Sturgeon (the American science fiction writer) in a 1957 article for *Venture* magazine and cited as such in the Oxford English Dictionary.

5. If you are behind one of these machines now, search for your terminal application using Spotlight and type `man wc` in the prompt (q to exit). For mere examples, see https://github.com/xpmethod/dhnotes/blob/master/command-line/109-text.md.

6. For more on the influence of Wikipedia, see Collier and Bear, and Callahan and Herring. It is a point made by Benjamin Mako Hill in his *Almost Wikipedia*. Another good summary comes from Garber, “The Contribution Conundrum.”

Bibliography


**API and Atom Feed**

Data about this text is available via read-only JSON API endpoints: sentences, annotations, comments, and index keywords.

Comments posted on this text can be followed by subscribing to this text’s Atom Feed.
We have been getting to work building our index for the menus data from the New York Public Library’s What’s on the Menu? project. We have traveled a long way from the idea that what the menus data set needs in order to be most useful to researchers is for someone to “clean it up.” As we discussed in our last essay, our plan instead is to create a data set of our own—an “index”—that links to and provides more information about the NYPL data.

We’ve had a provisional sense of what this index will be and how it will work for a while but, now that we’re actually building it, a few questions have arisen that we thought might be worth discussing publicly. The issues of most interest to us at this moment involve: what linked data techniques we should
use to relate our data to the NYPL data, how to use identifiers from the NYPL data to make explicit links between our data and theirs, and how the linked data we’re creating might later be used or queried. This is very much work in progress so please let us know what you think—you can comment on any paragraph.

**Index**

Our index is a set of hierarchically-organized concepts about the domain of food. The first version of this index took the form of lists, posted to a wiki which we could collaboratively edit. For the next version, we wanted the index data to be more machine-processable. The [Simple Knowledge Organization System (SKOS)](http://www.w3.org/2004/02/skos/) standard maintained by the World Wide Web Consortium (W3C) is designed for just the use case we have imagined. SKOS supports web publication of “the basic structure and content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, folksonomies, and other similar types of controlled vocabulary.”

SKOS models concepts, their names, and their relationships to each other. With this vocabulary we were able to
take the notes from our wiki and (with the help of Ed Summers) express the structure we wanted using the elements from SKOS. As a way of sketching out how this would look, we decided to serialize our data as **JSON-LD**. (We don’t expect to be writing out the serialization of our data by hand, but to work through some examples, this format seemed readable enough to be useful):

```
{
  "@context": {
    "@vocab": "http://www.w3.org/2004/02/skos/core#",
    "title": "http://purl.org/dc/terms/title",
    "creator": "http://purl.org/dc/terms/creator",
    "@base": "http://www.publicfare.org/def/food/"
  },
  "@id": "",
  "@type": "ConceptScheme",
  "title": "Public Fare Food Terms",
  "creator": [
    "https://github.com/trevormunoz",
    "https://github.com/katierawson"
  ],
  "hasTopConcept": [ {
    "@id": "Food",
    "@type": "Concept",
    "prefLabel": "Foods",
    "narrower": [ {
      "@id": "Animal",
      "@type": "Concept",
      "prefLabel": "Animal Foods",
```
"scopeNote": "Used to designate the type of animal that is the source of the food."

"narrower": [{
    "@id": "Meat",
    "@type": "Concept",
    "prefLabel": "Meat"
  }, {
    "@id": "Poultry",
    "@type": "Concept",
    "prefLabel": "Poultry"
  }, {
    "@id": "Seafood",
    "@type": "Concept",
    "prefLabel": "Seafood"
}]

}, {
    "@id": "Plant",
    "@type": "Concept",
    "prefLabel": "Plant Foods",
    "narrower": [{
        "@id": "Vegetable",
        "@type": "Concept",
        "prefLabel": "Vegetables",
        "narrower": [{
            "@id": "Arugula",
            "@type": "Concept",
            "prefLabel": "Arugula",
            "altLabel": "Eruca sativa"
        }, {
            "@id": "BrusselsSprouts",
            "@type": "Concept",
            "prefLabel": "Brussels sprouts",
            "altLabel": "Brassica oleracea (gemmafera)"
        }, {
            "@id": "Celery",
            "@type": "Concept",
            "prefLabel": "Celery"
        }]
    }, {
        "@id": "Fruit",
        "@type": "Concept",
        "prefLabel": "Fruit"
    }]
}
While modeling a domain as large and complex as “food” might be (is definitely) a sisyphian task, this does not negate the value of modeling as a data curation activity. For our project, we are beginning with just one corner
of the food domain—a corner that maps to “dining out” food practices, primarily American, primarily from the twentieth century. We are purposely developing this model from the “top down”—starting with “Food” then broad categories of food like “Meat,” “Vegetables,” etc.—because such a domain model provides a level of abstraction that we think people will be able to use to more easily see comparisons, trend lines, relevant importance, and certain types of outliers. We are developing this conceptual structure not based on specific terms contained in the NYPL data but instead based on our own judgement about what would effectively serve potential users. We hope our model of the domain will also provide the groundwork for researchers to subset and refine the data for their own purposes. By linking a domain model, even a simplistic one, to the dataset from NYPL we can enable uses beyond what is currently possible. The NYPL dataset aggregates transcriptions of representations printed on historic menus. At bottom this data is a set of letters presented in order for a customer to say, “I want this” (or a banquet attendee to know “we will be served that”). Inevitably this produces a great deal of variation until the only effective way to subset the
data is by date of the menu (which was carefully recorded when the object was collected by the Buttolph or other librarians). The NYPL dataset only indirectly provides information about foodstuffs; we believe based on these texts that the food was present (at least waiting in the kitchen pantry).

The practical challenge of adding a domain model to data which we did not create meant figuring out how to articulate the relationships between our SKOS concepts and NYPL’s data.

**Explicitness**

Specifically, we spent time trying to figure out the “proper” linked data way to accomplish this (and thought more than once that we had solutions when we didn’t). We had to reason our way—statement by statement—through the ways that our set was linked to NYPL’s in order to find what we wanted to say (and also what we didn’t.) Our linked data graph will include assertions like these (first in English):

1. A thing identified by a URI from the NYPL menus domain—for example
   <api.menus.nypl.org/dishes/dishes/8371>—is some data
2. We have defined a concept (carrot as foodstuff) and given that concept a URI—for example, `<www.publicfare.org/def/Carrot>`

3. We assert that the subject of the data identified in #1 is the concept in #2

Importantly, this set of statements says only that the thing from NYPL is a little chunk of data—a series of key/value pairs. The identifier is minted based on the uniqueness of the string found in the key called “name.” We don’t believe the thing identified by the URI has any other ontological status—while the NYPL application calls these “dishes”—it does not treat them as things. If we believed that this data modeled dishes-as-things, you would be able to tell the difference between being served a dish of “carrots” and a dish of “Carrots.” These so-called “dishes” are (changing) aggregations of the appearances of certain strings in the NYPL’s application database based on the activity of volunteer transcribers.

In our formal RDF model, we use a couple of additional statements to capture our understanding of this situation in an explicit way:
@prefix dc: <http://purl.org/dc/terms/> .
@prefix dcmitype: <http://purl.org/dc/dcmitype/> .
@prefix foaf: <http://xmlns.com/foaf/spec/#term_focus/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

<http://api.menus.nypl.org/dishes/dishes/8371> a dcmitype:Dataset;

<http://www.publicfare.org/def/food/Carrot> a skos:Concept;

**Pointing**

While the discussion above suggests that we knew what URI to choose to link to, this was yet another decision we had to make. Given how the NYPL data can be accessed, we saw three options:

1. Use the URI for the web page for the dish Pros: accessible, takes people back to the source, including to images of the menus; maintained by NYPL Cons: an odd landing point that doesn’t have clearly structured data; doesn’t reveal as much data easily as the API
2. Use the URI that will return the JSON document for the dish from the NYPL API Pros: nice structured data, including multiple points into
the content and information about it; maintained by NYPL Cons: needs a key to access so not as user friendly

3. Republish the data from the API within our own JSON documents
   Pros: includes all the data of the NYPL API, but is accessible Cons: means we have to store and maintain that much more data

Initially we hesitated to link to the URI for the JSON because the data about this URI could not be directly accessed by other users if they did not have a key for the NYPL API. However, this does not seem to deter other creators of linked data and is the most precise way to refer to what we want to talk about. Further, we realized that we could use predicates from the Europeana data model to have it both ways. We will point to both the JSON document from the NYPL API and the NYPL dish web page, and these two entry points are valuable to other researchers because they express different information.

**Queries**

We set out to build a small application that would allow us to aggregate dishes that would have the same value for “name” if only differences in spelling,
word order, and pluralization were normalized. However, now that we are beginning to index the data with concepts, this seems a more promising strategy than the (more time-intensive) project of aggregating and choosing normalized forms for similar strings. At this point, we’d rather scholars (including ourselves) be able to query our linked data for dishes that have “carrot” and “soup” as subject than to find matches for the specific string “carrot soup” in the data. This would allow them to get “carrot soup,” “chilled carrot vichyssoise,” and “soup of carrot and leek”—which is to say, direct experience of the diversity of the data.

We are both attending the Digital Library Federation Forum this week and look forward to talking with our colleagues about data and decisions.
The Point of Collection

The conceptual, practical, and ethical issues surrounding “big data” and data in general begin at the very moment of data collection. Particularly when the data concern people, not enough attention is paid to the realities entangled within that significant moment and spreading out from it.

I try to do some disentangling here, through five theses around data collection—points that are worth remembering, communicating, thinking about, dwelling on, and keeping in mind, if you have anything to do with data on a daily basis (read: all of us) and want to do data responsibly.

1. Data sets are the results of their means of collection.

It’s easy to forget that the people collecting a data set, and how they choose to do it, directly determines the data set.

An illustrative example can be found in the statistics for how many hate crimes were committed in the United States in 2012. According to the
FBI Uniform Crime Reporting Program (UCR), the number was 5,796. However, the Department of Justice's Bureau of Statistics reported 293,800 hate crimes.

The reason for the variation was simple. The URC gathers data that is voluntarily reported by law enforcement agencies across the country. The Bureau of Statistics, on the other hand, distributes the National Crime Victimization Survey, which collects data from the victims of hate crimes. The result is a more transparent and inclusive surveying.

Same data set, two different means of collection, two wildly different results. What they show is an important fact we must keep in mind: There’s no pure objectivity encoded into data sets. Each one is the result of a number of human processes and decisions that affect, in a variety of ways, the data that they aim to report. In this sense, the moment of data collection starts before any data is actually produced.

2. As we collect more data, we prioritize things that fit patterns of collection.

Or as Rob Kitchin and Martin Dodge say in *Code/Space*, “The effect of abstracting the world is that the world starts to structure itself in the image of the capta and the code.” Data emerges from a world that is increasingly software-mediated, and software thrives on abstraction. It flattens out individual variations in favor of types and models.

As we abstract the world, we prioritize abstractions of the world. The more we look to data to answer our big questions (in areas like policing, safety, and security), the more incentives we have to shape the world into an input that fits into an algorithm. Our need to generate things that feed a model rings true even in cases where the messy bounds of experiences can’t be neatly categorized into bits and bytes, or easily retrieved from tables through queries.

Biometric data is a great example of this. Fingerprint authentication technologies and iris scanners point to a system where individuals are uniquely identified through metrics and data. In order for this to work, people themselves have to be conceptualized more and more as machine-readable.

3. Data sets outlive the rationale for their collection.
Spotify can come up with a list of reasons why having access to users’ photos, locations, microphones, and contact lists can improve the music streaming experience. But the reasons why they decide these forms of data might be useful can be less important than the fact that they have the data itself. This is because the needs or desires influencing the decisions to collect some type of data often eventually disappear, while the data produced as a result of those decisions have the potential to live for much longer. The data are capable of shifting and changing according to specific cultural contexts and to play different roles than what they might have initially been intended for.

Ultimately, the question of intention behind the collection or generation of a data set can be rendered irrelevant. Thinking through the moment of collection can reveal the distance between it and the data’s use. And it’s often far more critical to think about the potentials and possibilities surrounding what can be done with collected data.

4. Corollary: Especially combined, data sets reveal far more than intended.

We sometimes fail to realize that data sets, both on their own and combined with others, can be used to do far more than what they were originally intended for. You can make inferences from one data set that result in conclusions in completely different realms. Facebook, by having huge amounts of data on people and their networks, could make reasonable hypotheses regarding people’s sexual orientations.

People who work with data know this intimately, but it can often be difficult to see the connections between the collection of one thing and the inference of something else. Unfortunately, the effects of these connections can become very strongly felt. As Bruce Schneier puts it, “data we’re willing to share can imply conclusions that we don’t want to share.”

5. Data collection is a transaction that is the result of an invisible relationship.

This is a frame—connected to my first point—useful for understanding how to think about data collection on the whole:

Every data set involving people implies subjects and objects, those who collect and those who make up the collected. It is imperative to
remember that on both sides we have human beings. I point this out not for any fluffy reasons related to humanism or human-centered design, but because power arises out of hierarchies, interactions, and dynamics. The below-the-surface work of a particular data set is joined to the reasons and means that created it and the relationships running through those reasons and means. If we can keep that in mind, we’re better positioned to see data as an intermediate result, one piece in a larger process, something that is as much human-oriented as it is systematic. The challenge is for us to keep in mind both aspects of data collection, to see systematic as well as human tensions and biases.

The point of data collection is a unique site for unpacking change, abuse, unfairness, bias, and potential. We can’t talk about responsible data without talking about the moment when data becomes data.

Points: “The Point of Collection” takes off from Mimi Onuoha’s recent Machine Eatable talk: Data as Process. Thinking through the implications of the moment of data collection, she offers a compact set of reminders for those who work with and think about data. Tape it to your monitor.—Ed.

Artist and researcher Mimi Onuoha is a Data & Society fellow and adjunct faculty at NYU’s ITP. Her work involves using data and code to explore new forms of storytelling, social critique, and interaction. She’s currently working on projects exploring missing data sets.
This article conveys the experience of designing, creating and analysing a Microsoft Access Database application in support of research into events in the lives of youth convicts, transported to Australia between 1826 and 1837, as reconstructed from the surviving institutional records. The source is an unpublished assignment submitted by the author for the Advanced Diploma in Local History, Department of Continuing Education, University of Oxford, 2013.
The early nineteenth century was a period that saw the displacement of rural labour, the loss of youth apprenticeships and a rapid increase in urban populations of over 40 per cent; predominantly comprised of the young. As a consequence, with employment scarce, juveniles formed a large proportion of the criminal class. At the same time, from the 1820s, the impact of Peel's law reforms removed the primacy of the death penalty even though the numbers of those capitally convicted rose three-fold from 1825 to 1835: and, transportation remained the option for those considered to be serious offenders, which was equally applied to the young, those under sixteen. Public attitudes equally evoked anguish at the state of the ‘soul of the little ragged urchin’ and fear of a ‘hostile power’ and this debate, with its legislative consequences, shaped youths’ experience of the criminal justice system. This article is concerned with investigating this experience for a number of youths, recorded in the proceedings of the Old Bailey Online for trials between 1822 and 1836, many of whom were held on a convict prison hulk for boys, from which they were transferred for transportation to Australia. The approach taken is to set the historical context for the crimes selected and design a database that uses institutional sources to record details of each individual in their journey through the criminal justice system; while, at the same time providing aggregate data about cohorts, in answer to the research question ‘what can be understood about the lives of youths in the criminal justice system from the institutional records of their time’. A number of approaches to table and query design, in a relational database, are considered in support of this dual goal: and, the preferred table structure and query design tested in terms of analyses for the research question. A conclusion is drawn as to the value of this application as a tool for historical research. The majority of youths transported to South-Eastern Australia were for crimes of larceny, in the hope that reform would arise from this punishment, and has left a trail of institutional records that can be used to analyse the research question. Contemporary stereotypes of juvenile criminals
fed on the Proceedings of ‘serious’ crimes, at the Old Bailey, including pickpocketing, burglary and crimes of violence: and, for those under sixteen who were transported to Van Diemen’s Land (Tasmania) and New South Wales between 1787 and 1852, some 94 per cent were for crimes of larceny. Transportation offered more than just removal in the hope that a boy would work off his assignment and gain a ‘ticket-of-leave’ only to set out anew in a new land. From 1824, hulks, such as the Euryalus, were set aside for youths in a separation from adults that sought reform but that, through criminal ‘contamination’, failed. A more directed regime of purely youth reform was instituted at Point Puer, in Tasmania, in 1834 which used separation, religious instruction and silence as well as discipline and learning a trade: with over 3,000 boys put through its fifteen years of operation. These reforms and institutions have left a trail of records which offer a rich source with which to study the experience of the individuals and cohorts. Eighteen individuals were identified from the Online Proceedings of the Old Bailey (OBP), from 1822 to 1836, which provides partial transcripts for many of the trials for serious crimes, committed in London and Middlesex, and which by 1778 became a formal basis for reports to the King concerning whether the convicts should be pardoned or executed; and, which is available online (table one). Consistent with its scholastic use, the OBP was a starting point from which original records or images were then sourced from The National Archives, the State Library in Tasmania and New South Wales in order to build a database in support of the research question. However, as for many historians, the topic of research is selected but the precise purpose to which the data will be put is not entirely clear: and, as a consequence the table design of a database application must offer flexibility at the cost of simplicity.
Defendants Sourced from the OBP Online

The research question requires a table design that is method-oriented but also supports a breath of analysis, and hence is organised around four entities and their relationships. Denley’s task comparison between model-oriented and source-oriented table design is used to profile this project which: is not prescriptive of question nor seeks a particular outcome but is oriented to learning; accepts an abstraction of data with moderate retention of sources and original text; balances a quantitative view of cohort data and an events view of an individual; is time pressured with results expected; and, readily accepts the use of ‘mainstream’ tools (table two). Table 2: Research Project Approach: Task Parameters

<table>
<thead>
<tr>
<th>Model Oriented design</th>
<th>H/M/L</th>
<th>Source Oriented Design</th>
<th>H/M/L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific research questions</td>
<td>M</td>
<td>Integrity of source is a high priority</td>
<td>M</td>
</tr>
<tr>
<td>Answers matter, not serendipity</td>
<td>M</td>
<td>Ensure use for those who follow</td>
<td>L</td>
</tr>
<tr>
<td>Data is regular and simple in form</td>
<td>M</td>
<td>Query both data and text</td>
<td>M</td>
</tr>
<tr>
<td>Abstraction, arbitrary decisions on data OK</td>
<td>M</td>
<td>Investment in time and effort</td>
<td>M</td>
</tr>
<tr>
<td>Quantitative analysis primary</td>
<td>L</td>
<td>Prepared to develop a bespoke system</td>
<td>L</td>
</tr>
<tr>
<td>Tight research schedule, results demanded</td>
<td>H</td>
<td>Prepared to learn an unorthodox tool</td>
<td>L</td>
</tr>
<tr>
<td>Use of ‘mainstream’ tools</td>
<td>H</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For the detailed table design, the PBE is preferred which adopts entities for person, event as ‘factoid’, location and source: and, where a factoid is an assertion that accepts plurality of interpretation, while avoiding the Computer Science meaning of entity as object, both of post-modernist concern (figure one). Similarly, this initial design will adopt the entities: Person, Events, Locations and Sources; for which, an event occurs at a location on a date, asserting an association with a
person; the data for which is abstracted from a source; and, these tables will be tested using institutional events associated with the youths transported, to derive the final design.

Figure 1: The table Design of the PBE database

The final table design confirmed person for whom institutional events are asserted to occur at a location on a date with relationships that support queries of this, as well as, aggregate views of events for cohorts undergoing similar or different experiences. There are four types of events that each differ in their source data structures, namely: Bio details, now acknowledged as an event as it is date bound, as are Transportation, Trials and Other events; in, a design that allows for specialist input from specific sources and querying within a table, such as for height, by age, of youths tried the Old Bailey. Each person, is the focus of the research and experiences many events, on dates, at locations in a one-to-many relationship, for events that occur at locations in a many-to-one relationship; and, without any many-to-many relationships there is no need for a junction table (figure two). Source, an entity in the initial design, is now incorporated within the event and location tables for ease of reference.
This design preserves a high degree of flexibility: in, the potential addition of new event types, such as census; and, supports the potential to scale up the project for all youths, in the OBP, who were transported to Point Puer or other locations. The issue of assertion is also supported by the opportunity to enter contrary views as other event types, such as in the interviews of youths by W.A. Miles, ‘an uneasy moral entrepreneur’, on the Euryalus in 1835 or the assertion by Cameron Nunn that Samuel Holmes was the source for the ‘Artful Dodger’. All relationships create a record linkage between Person, Event and Location and this should support the simplification of data entry with some limitation of error, such as in the use of a location table; and, ensure that the results of queries are more accurate as the result of a deliberate selection of entity and attributes. Date is an
issue, as in the interviews on the Euryalus in 1835, when only the year is recorded, requires a judgement over the nearest day and month of the event; also, events such as transportation, with a start and end date are treated as two different events for the purpose of record design. The records for Charles Downes included a variant ‘Downs’ which was verified as a synonym through record linkage across four different sources. Each table has a separate input form as the sources are quite distinct and lend themselves to project specialism. The person table conforms to the requirements of Third Normal Form (3NF) but for event tables location, name and source are deliberately retained to assist the historian: and, for scaling up, coding lists could be used and 3NF attained. Through trial and error, eight queries demonstrated learning in the use of SQL while, at the same time progressing enquiry across the research domain, for both the individual and the cohort. In figure three, the bio details sub query simply selects the biological details of those youths transported and recorded on arrival in Australia: and then modified, through selection criteria, to derive a list of those who were on the youth hulk ‘Euryalus’ or at Point Puer for whom the other events, post-trial, were later extracted to provide a cohort experience, held, to a degree, in common. The trial details provided a comparative view of the cohort and the value of all thefts were converted to decimal, through expression builder, and exported to MS Excel to derive an average value per guilty verdict. The recorded heights of a cohort were exported to Excel and graphed to provide a comparison with known research on the heights of London youth convicts. The issue of deriving a ‘biography’ for a selected person by query across all tables initially returned an error message “…ambiguous outer joins” which could be solved by repeatedly creating a query on one table that was then included in a subsequent query; although, it made more sense to collate this in the form of a report, compiled from sub-reports that extracted data on a particular event type; in an approach that
achieved compilation but at the loss of header information, a known limitation of MS Access reporting.

When populated with data from institutional records, the database supports the analysis and construction of a personal timeline as well as aggregate views of events occurring for cohorts that examine questions on the nature of penal reform in practice; and, through multiple source assertions supports contradiction and data verification. The power of the database to analyse and compile the events for a selected person is evident from the timeline constructed for Samuel Holmes which asserts: an early delinquency through to time on the youth prison hulk, Euryalus; with assignment to Point Puer, a youth penal reform prison, where he experienced punishments; and, later with a permission to marry recorded in 1847, possibly to Elizabeth Clark, both on ‘Tickets Of Leave’; and, by 1851, is free in occupation as a farmer, married with a female child recorded in the census (figure four).
Figure 4: The Institutional Biography of Samuel Holmes

Through record linking and querying, questions were raised, such as for Sarah Holmes, associated on the Tasmanian Convict Database with Samuel Holmes, who is not listed in this 1842 census record, for which there is no evidence of permission to marry; and similarly, of the assertion by Nunn, in 2011, that Holmes was the source for Charles Dickens’s ‘Artful Dodger’, whose description varies from that of the institutional source (figure five). Three valuable views of cohorts are immediately evident from the data. Firstly, the Mile’s interviews on board the Euryalus, in 1835, assert that most of this cohort sentenced to transportation for seemingly minor larcenies were already known to the court and hence the severity of the sentences when compared to the cohort average theft value of 67 pence.

Figure 5: The Artful Dodger and Samuel Holmes: A Source Comparison

Secondly, the height range of the cohort that arrived in Port Arthur, in 1836, is consistent with other findings that the average height of
London criminal youths was shorter than the norm with the criminals’ average at: 53.5” for a 13 year old; 59.1” for a 16 year old; and, up to 61.4” for 18 years old as depicted in figure six.

Figure 6: Cohort Heights: 1836 Arrivals at Point Puer

Thirdly, the establishment of Point Puer youth prison at Port Arthur, in Van Diemen’s Land, reflected a reform in penal policy and prison architecture with an expectation that through hard work, exposure to religion, education and vocational training youths would gain “a sense of worth, competence and even independence” (figure seven). This may well be the case, in that, despite evidence of regular punishments for Samuel Holmes and Charles Downes, while under a reform regime at Point Puer, the census of 1851 records Holmes as a farmer, married with a daughter and all three of Church of England faith, and not as dissenters which was one option on the census form; and for, James Edwards, under sentence of death in 1835, who is recorded as free in 1841.
In conclusion, this article conveys the experience of designing, creating and analysing a Microsoft Access Database application in support of research into events in the lives of youth convicts, transported to South-Eastern Australia, mainly for larceny, between 1826 and 1837, in a pursuit of the question ‘what can be understood about the lives of youths in the criminal justice system from the institutional records of their time’. Over this period, the impact of
Peel’s law reforms removed the primacy of the death penalty and, with public attitudes equally evoking anguish and fear, the ensuing debate, with its legislative consequences, shaped youths’ experience of the criminal justice system. Youth prison hulks were considered to have failed and a more directed regime of purely youth reform was instituted at Point Puer, in Tasmania, which used separation, religious instruction and silence as well as discipline and learning a trade for 3,000 boys over its fifteen years of operation. In support, a relational database development took a method-oriented approach within which a table design adopted the four entities of person, event, location and source: and, which through refinement attained 3NF, in part, while retaining flexibility for the historian in a degree of record attribute redundancy; in a final design that provided record linkage of multiple sources of events that occur at a location and a date in a person’s life. Relationships created across and within tables, provided for individual and aggregate analysis of events for selected youths undergoing similar or different experiences. Trial and error, determined the most useful set of queries that demonstrated learning, while at the same time progressing enquiry across the research domain, and defined the associated input forms and output reports. When populated with data from institutional records, the database supports the analysis and construction of a personal timeline as well as aggregate views of events occurring for cohorts that examined questions on: the nature of penal reform in practice; height ranges by age; aggregate data on theft values; and, record linking of multiple assertions that surfaced further questions and supported data verification. In these outcomes, the database has demonstrated a very clear value and the potential for a scaling up, for example, for the 3000 youths who experienced Point Puer, where records survive, and supports an examination of changes in prison reform, as well as offering potentially new lines of enquiry such as average heights, between dates, and the relationship to prison diet.

BIBLIOGRAPHY PRIMARY SOURCES Convict Applications to Marry CON
Using Old Bailey Records for Youth Convicts, transported 1826-1837 – Quainton Local Historian


Convict Arrivals in Tasmania 1836 HO 10/29 The National Archive


Transportee biography.accdb in ADLH_Turnbull_assignment5_dbase

SECONDARY SOURCES


INTERNET SOURCES
4/12/2018  

Using Old Bailey Records for Youth Convicts, transported 1826-1837 – Quainton Local Historian


Volume 6: Issue 2, pp.33-43 URL: https://docs.google.com accessed
23rd April 2013 Dickens, C (1866) The Adventures of Oliver Twist,
Chapman & Hall URL: http://books.google.co.uk accessed 23rd April
2013 Dow, G. and Brand, I (1986) “Cruel only to be kind” ? Arthur’s
Newman, T (2005) Becoming Tasmania: Renaming Van Diemen’s Land
Parliament of Tasmania, URL:
g.htm accessed 23rd April 2013 Gandevia, B. (1978) Tears Often
Shed: Child Health and Welfare in Australia from 1788 Pergamon
accessed 16th April 2013 History of the Metropolitan Police Force
http://www.met.police.uk/history/timeline1829-1849.htm accessed
identity: Understanding the pursuit of the individual in
Approaches and Applications. A Handbook, (Occasional Publication of
the Unit for Prosopographical Research Vol. 13, Oxford), URL:
http://prosopography.modhist.ox.ac.uk/course_syllabuses.htm
Normal Forms in Relational Database Theory", Communications of the
ACM 26(2), URL: http://www.bkent.net/Doc/simple5.htm accessed
doi:10.1371/journal.pone.0053285 accessed 27th April 2013 Magery,
S. (1978) The Invention of Juvenile Delinquency in Early Nineteenth-
Mathisen, R.W. (2007) Where are all the PDBs?: The Creation of
Prosopographical Databases for the Ancient and Medieval Worlds in

Identifying a Demographic Crisis through Baptism and Burial Records

Apprentice Records and the economic and social transition of Liverpool, 1650-1750
Introduction

The collection of historic menus at the New York Public Library (NYPL) was created by a woman named Frank E. Buttolph. From the beginning, she intended the collection to be a research collection. In much of what has been written and documented (and repeated over and over again) about the menu collection, commentators launch into discussions of the collection’s size and richness, pausing momentarily to regret the paucity of information available about its creator. According to this common trope,
Buttolph is a “mysterious and passionate figure”\(^1\) who led a “quite private life”\(^2\) about which (regrettably) not much can be known\(^3\). Yet in trying to make good on Buttolph’s original intent—that the collection be used for research—and also in trying to curate the digital data set that NYPL has created from this physical collection, we found ourselves unable to accept the historical absence of the collection’s creator.
We wanted to understand the provenance of the menus data and, therefore, needed to know how the physical collection that underpins the digital data came to be. To maintain, or even improve, “the interest and usefulness” of the menus data for humanities scholarship, we want to situate the collection in local and historical context. To achieve this, we needed to understand more about Buttolph herself. The more we researched Buttolph’s life, the more we found ourselves compelled to tell a better and fuller story about her and her work as a collector in the early twentieth century.

The details of Buttolph’s life are not as unknowable as we had been led to believe. A fairly sturdy outline can be assembled from surviving documentary sources—work made easier, no doubt, by the growth of digitized collections available for full-text searching. Yet having more of “the facts” about Buttolph at our disposal paradoxically increased the challenge of doing research with her collection. We were prepared to find that aspects of Buttolph’s personal history had perhaps overtly shaped which menus she collected—either through access (to particular strata of society versus...
others) or through enthusiasm (for particular types of menus). To some extent, this is the case, but the interconnections between the life and the collection—and now its descendant, the data set—are less overt and more interesting.

Our research pushes back against the most oft-repeated stories about the historic menu collection and its creator. These brief narratives emphasize the eccentricity of the menu collecting project or frame it as a personal obsession, and present Buttolph as a lucky amateur and an enigmatic spinster. Instead of the personal project of an amateur collector, which happened, by good fortune, to become folded into the collections of the New York Public Library, we now understand this collection as a research collection built by a woman invested in cultural preservation and public knowledge-building. We can see the menus collection as the contribution of a woman whose work was in sync with the project of the larger institution around her during one of the library’s most diverse and active periods of collection building. Indeed, finding out more about Buttolph suggests that what is mysterious here is not her biography but the prevailing
representation that casts her life as mysterious and not profitable or interesting to investigate further.

Once we focused on this more interesting story, we had to grapple self-consciously with the challenges of how to make sense of this woman, her work, and the data that derives from that work. As we researched Buttolph, we found ourselves asking questions about what it meant to tell her story. We faced what historian Jill Lepore calls the “tricky work” of writing about people’s lives. In the case of Buttolph’s story, the challenges stem from records, which (while not so barren as suggested) must still be counted slender and from the awkward fit between the picture of Buttolph we can develop and traditional “Great Man” or even Great Woman notions of biographical subjects.

In her essay in The Journal of American History, Lepore makes a distinction between different types of writing about people’s lives based on the aims of each type. Writers can “seek to profile an individual and recapitulate a life story,” in order that readers might know the person better. Or, they can “address themselves to solving small mysteries about a person’s life as a
means to exploring the culture.”

Researching Buttolph’s life had the built-in dramatic interest that motivates not only scholars but also genealogists and family historians—“tracing [an] elusive character through slender records,” celebrating each new crumb or cache of documentary evidence. Every discovery of a detail about Buttolph seemed to invest us in a project of recovering her personal story. Yet what should we make of one (incomplete) life story?

We want to understand Buttolph’s life as a way of better curating the data set built from her menu collection. When placed in historical context, the tension between exploring Buttolph’s life story for itself or for what it can tell us carries a specific charge. We suspect that the lack of attention paid to Buttolph’s life comes not only from the specific circumstances of that life but also from the wider culture in which she lived (and in which we still live). That is to say: it matters that Buttolph was a woman.

We tell her story in the context of historians who have “writ[ten] women into librarianship.” According to Suzanne Hildenbrand, as summarized by Phyllis Dain, these approaches
include: 1) “compensatory history that rescues women from obscurity;” 2) “delineation of a separate women’s culture with its own validity;” and 3) “the study of women in history, interacting with men and involved in the dominant patriarchal culture.”

The thrill of the historical chase, the excitement of “discovery”, the pleasing sense of fairness might incline us toward a project of rescuing Buttolph, the individual, from obscurity. However, since our project aims to use the menu data to produce new knowledge, the third kind of analysis is most salient: acknowledging Buttolph’s gender as a significant factor helps us to account for “the systematic ways in which sex differences cut through society and culture and [confer] inequality upon women.”

The interpretive strategies that comprise humanities research, which attend to dynamics of culture and history, give us purchase on questions just like these. Thus, attention to Buttolph’s life and its representation is part of the work of humanities data curation.

If the common, easy narratives about Buttolph and the collection are problematic not just for what they get wrong about the woman but also for what they get wrong about the
meaning of the resulting collection and data set, what considerations would improve them? As a beginning, we reconsider her collecting practices, her relationship to the library around her, and her gender as ways to critique the presentation of a “mysterious and passionate Miss Frank E. Buttolph.”

**When Women Acquire**

When women acquire the collecting mania, they often branch out in the most unexpected directions, and take up what appear to be trivial and foolish specialties. Sometimes these develop into collections of real value, the more so because no one else thought it worthwhile to make such collections.

It was a fairy-tale queen, was it not, who was so wrapped up in her collection of buttonholes? We thought her very silly in the fairy-tale, but we appreciate her now as the patron saint of all museums of domestic art.
When Miss Frank Buttolph began to collect menu cards and bills of fare, no one thought of it as anything better than a rather tiresome freak, on which a vast amount of energy was being wasted that might have been expended better.\textsuperscript{11}

The writer of the \textit{The Literary Collector} magazine’s March 1905 “Notes” section presents “Miss Frank Buttolph” as an amateur, eccentric, a woman obsessed with a bizarre collection. This representation is not unusual, for Buttolph or many other woman collectors of the time.\textsuperscript{12} Over the course of this brief notice, \textit{The Literary Collector}’s anonymous writer seems to be won over to the value of Buttolph’s collection. “When she began to get menus in all languages from all countries, [the collection] grew a bit interesting to the scoffer,” the writer avers. He or she concedes that the collection will be “invaluable to the late twentieth or twenty-first century writer of the history of the nineteenth-century civilization,” without fully subscribing to this conclusion or recanting the gendered caricature which opened the piece. He ends by framing Buttolph’s aims using a turn-of-phrase that may undermine them: “For
the historian of the future has Miss Buttolph consciously been making this collection, and she feels sure of his appreciation.” In many ways, this note in The Literary Collector encapsulates how Buttolph and her work have been viewed. The depiction’s acerbic nature makes plain stereotypes that surround Buttolph, often subtly, into the twenty-first century.

Menu from the Columbia Restaurant, 1901, from the Buttolph Collection of Menus, New York Public Library.

By telling the story of Frank E. Buttolph’s collection—yet again—and telling the story of those stories told
about her, we hope to provide new contexts for understanding her work as a collector. We begin with stories about the origin of the collection. When Buttolph and her collection were featured in newspaper articles, journalists often retold a story about how Buttolph’s awareness of the changing century—specifically her first sight of the year “1900” printed on the menu of a restaurant on New Year’s Day—inspired her project of collecting for the library. Buttolph herself recounted this story in a letter to the New York Public Library’s board of trustees:

On New Year’s Day I stopped in the Columbia Restaurant for lunch and thought it might be interesting to file a bill of fare at the library. A week later the thought occurred [sic], why not preserve others? As a result 930 have passed through my fingers to the Astor Library.¹³

The first notice of the collection, from The New York Times might suggest that Buttolph was just another donor among what was, at the turn of the twentieth century, a growing roll of august and high-minded contributors to the new and expanding New York
Public Library. “Gift for the Public Library,” from February 15, 1900, reports on a donation of new funds for purchasing materials on sociology and economics and mentions that “Miss Frank E. Buttolph offered to give to the library a collection of a thousand hotel and restaurant menus. She has already turned in over 900 ....” It would seem that, in the writer’s mind at least, Buttolph’s project could be classed with the gifts and donations as part of ordinary library business. Yet, very quickly the details of the story do not support the view of Buttolph as a traditional donor.

A later New York Tribune retelling of the story of the New Year’s menu that inspired the collection tells a different story. The anonymous (likely male) author’s assumptions revert gender roles in a way that robs Buttolph of agency and thus normalizes her project as ordinary work of the library:

Her first effort was due to seeing for the first time on the bill of fare of a restaurant in Union Square the date 1900. She took away the bill with her and spoke to her superiors at the Astor Library about it. They proposed keeping it, and
Miss Buttolph at once decided to begin collecting bills of fare.  

Giving the authority for the project to the library buttressed male authority since even though, by this time, most librarians were women, library administrators and leaders were almost all men. One of the journalists from The New York Times who covered the collection likewise reported of Buttolph’s “peculiar interest” that “Dr. Billings, the librarian, has enrolled it amid the archives of learning.”

Comparing other contemporary newspaper articles about the menu collection, many of which feature reported statements by Buttolph herself, suggests that she did not mind changing the tale of the collection’s creation to make a better story. Exactly when, how, and why the collection was started changes between tellings. The versions allow us to see the effects of gender and power relations shaping the narrative and how Buttolph used press attention in service of her project. The New Year’s story has dramatic appeal, but was the collection project started in early 1900, or had Buttolph already begun collecting? Did she “walk into the library and ask to speak to the director” about commencing the project, or was
The library's institutional records add little information about the collection's genesis. The shape of the story retold in the brief accession sheet accompanying the "Miss Frank E. Buttolph Papers" is familiar though the details differ slightly from the newspaper accounts. The library’s record recounts that Buttolph approached the library around the turn of the twentieth century; NYPL Director John Shaw Billings agreed to preserve menus and engaged her to collect them for the library, which she did until the end of her life. The exact phrasing is interesting in its elliptical treatment of just how the commencement of the project was negotiated: "Miss Buttolph inquired whether the library would preserve some then current menus." There is nothing particularly authoritative about the accession sheet, which is dated March 30, 1987, and represents little more than a convenience for researchers, appended to Buttolph’s personal papers almost sixty-five years after their acquisition as part of the Library’s formal process of registering the collection. Indeed, this record, which asserts only that Buttolph "had
friendly relations with the New York Public Library for many years” appears to have been cut-and-pasted together. The bland formulation about “friendly relations” was literally copied into the finding aid for Buttolph’s papers from a notice written by a library staff member on the occasion of Buttolph’s death in 1924.

Buttolph was mentioned only twice in the library’s own regularly-published bulletins and annual reports. The first (and most extensive) mention, in The Bulletin of the New York Public Library published in January 1907, is a description of her collection included in a summary of the previous half-year’s activities of the Reference Department. This notice refers to “the collection of menus, formed largely by Miss Frank E. Buttolph” as though it were part of the library’s collection and not an outside gift and without giving any indication that Buttolph was not a member of the library staff. However, when Buttolph’s collection is mentioned again, in the Director’s annual report (for calendar year 1909), it is a much briefer mention in a section on “Purchases and Gifts.” The difference may be simple inconsistency—it was surely difficult to keep straight the details of various collections year after year in reports.
with multiple authors. Yet if we combine this discrepancy with the oddities of phrasing in Buttolph’s letter (her reference to “fil[ing] a bill of fare at the library”), in The New York Times reporter’s story (that Buttolph had “turned in” over nine-hundred menus), and in the library’s surviving documentary record of the collection’s origin, they seem to suggest a more unusual arrangement than donor and gift. The slipperiness of details about how the collection started reflects the ambiguity of Buttolph’s position.

“Is there not some place in which my knowledge ... could be utilized?”

At the end of January 1900, about two weeks before her proposal to the library trustees to make a “donation” of the menu collection, Buttolph wrote to NYPL Director John Billings to inquire about the possibility of employment. “I see a great many women employed at the Astor Library. Is there not some place in which my knowledge of books, countries and languages could be utilized in a manner remunerative to myself?”[17] We have not been able to find any record of Billings’ reply (if he made one) and Buttolph never did gain
a professional position at the library. However, this brief application does introduce a glimpse into her biography at the moment she was beginning her most active association with NYPL.

Women working in the New York Public Library at 115th Street, from the New York Public Library.

One significant distinction between Buttolph and the “great many” women employed at the library may have been her age. When she made her request, Buttolph was fifty-six years old (only six years younger than Billings himself). Perhaps more significant than her age is the fact that she seems interested in employment (not volunteer work) for the library. Given the very existence of the named collection and her long-running position as a volunteer, it would be easy to assume that Buttolph had a different class background and position than she did. However, her biography draws into relief her
collecting work as part of a longer investment and career in education.

Frank E. Buttolph was born Frances Editha Buttles in 1844, in Mansfield, Pennsylvania. Due to the presence of a state normal school, this small town in north central Pennsylvania close to the border with New York state offered an important educational opportunity to the young Buttolph. The normal school replaced an earlier “classical seminary” in Mansfield in 1862, and offered professional training to prepare high school graduates for careers as schoolteachers. Buttolph was a member of the first graduating class of this new institution. Indeed it was the alumni reports in the annual catalogue/yearbook of the Pennsylvania State Normal School, Mansfield, Tioga County, now digitized and made available at the Internet Archive, that helped us follow Buttolph’s career before she came to NYPL.

After her graduation, Buttolph embarked on the career of an itinerant school teacher, usually spending only a year in each place. Based on the Normal School catalogs, we can follow her from Mansfield to Rahway, New Jersey; Cattletsburg, Kentucky; Wilmington, Delaware; Brooklyn, New
York; and Tivoli, New York. Buttolph
and her younger sister Permelia (also a
Normal School graduate) ventured as
far west as St. Paul, Minnesota, in the
mid-1870s. She taught a range of
subjects and, as is evident in her later
collection development and translation
work, was versed in several European
languages.

Many of her teaching posts were at
schools associated with the Episcopal
Church, and Buttolph appears to have
been an active church member
throughout her life. In the early 1880s,
she traveled to Germany and may have
visited other destinations in Europe.
Gradually, Buttolph’s travels gravitated
toward to New York City. Though the
evidence becomes spotty, she seems to
have more or less relocated there by
1887, with the exception of a return to
Mansfield around the time of her
parents’ deaths in 1890 and 1893
respectively.

From various sources, it’s apparent that
Buttolph lived in the same general
section of Manhattan, between 14th
and 19th streets, never far from the
Astor Library, for the entire twenty to
thirty years she remained in New York.
She did gradually move from the east
side (near what is now Beth Israel
Hospital) to the west side near the 9th Avenue elevated railway, to a section of Chelsea in which the Depression-era New York City Guide noted, “a large number of brownstones, originally built as private residences, have been converted into lodginghouses.” For an educated, well-traveled, single woman, the Astor Library would have been an attractive local resource and perhaps a natural place to spend her free time. While it appears she may have been collecting before her formal relationship with the Astor Library began, the details of her engagement with cultural preservation emerges most clearly through the trails of letters in periodicals and in archival materials from the NYPL.

**Buttolph’s Collecting Practices**

When reporters wrote about Buttolph’s collection over the next twenty years, they usually focused on its oddness and its celebrity interest—particularly in the form of royal and political menus. In aggregate, these articles seem to frame the collection as frivolous, despite the occasional lip-service to “history” and “posterity.” As a collector, they often present Buttolph as both the driving force in the collection and as a bit
eccentric. A quote from 1906 article suggests some of the discourse that surrounded the menu collection:

It cannot make its impression a part of literature, nor can it be described as an appendix to history, nor has it any place amid ‘old manuscripts.’ It must forever stand for what it is, the ‘Buttolph Collection,’ or, to describe it more elaborately, in the manner of old-time subtitles, it is ‘the feminine instinct for accumulation verified by a lady, with neatness, elegance, and artistic verisimilitude.’

Yet when examined closely, Buttolph’s collecting practices and principles identify her as someone committed to research and preservation, invested in producing a broad, high-quality collection. Firm distinctions between “amateurs” and “professionals” are not possible to make during this period. To assess Buttolph’s project, we need to look closely at the work she did without relying on labels such as amateur and professional. A close reading of even the heavily stereotyped stories about Buttolph often reveals details of her work. For example, in “Literature of Eating, Collection of Menus in the New...
York Public Library,” the author says, “When Miss Buttolph attempted to transcribe the bill the last line worried her greatly.” While the author goes on to tell a story that showcases Buttolph’s sense of humor, this detail also reveals that she was transcribing materials. She created notes about the provenance and content of many menus and designed an organizational schemes for cataloging the menus, which she implemented in the form of a card catalog, and she attended to the physical housing and preservation of materials.

Menu from the fifteenth anniversary banquet held by the Stewards’ Association of New York at Reisenweber’s Circle Hotel, Eighth Avenue and 58th Street, 1905, from the Frank E. Buttolph Menu Collection, New York Public Library.
The collection stamps that emboss almost every menu in the Buttolph collection, sometimes multiple times, may suggest the possessiveness that has been ascribed to collectors, most famously in Baudrillard’s “The System of Collecting.” Buttolph was part of a segment of turn-of-the-century collectors who were invested in preservation as an act of social engagement. 21 To get a truer measure of Buttolph’s collecting activities, we examine her project activities in ways promoted by Anthony Shelton, who says, “Too much recent opinion has tried to reduce and generalise the ‘collector’ to a common type person without examining the complex motivations underlying their actions or the relationships between their collections and their other productions and their worldview.” 22 To understand Buttolph’s other productions and worldview, we begin not with the menus, but with church bulletins.

In 1900, The Churchman, a weekly magazine for the Episcopal community, published a short piece in response to a letter Buttolph (under the name Buttles) wrote to the Bishop of the New York. In it, she expressed concern over “the woeful waste of artistic programmes of Church festivals.” She
had come to an agreement with sextons of two churches to take some their extras and distribute them to a range of people, including “the man at the umbrella stand in the Astor Library.” She goes on to encourage “a committee in each parish who would rescue these leaflets from the flame and then scatter them where there is genuine appreciation of their beauty.”

A few things in this short narrative are significant for understanding Buttolph’s relationship to collecting and the menu collection. First, she venerated the ephemeral. Second, she saw the aesthetic and cultural purpose of ephemera. Third, she takes the initiative to work with people within institutions to collect and share these materials. And finally, she had an idea of audience that is as broad as her idea of what is valuable—extending to the man at the umbrella stand.

Buttolph saw institutional collecting as a way of having ephemeral cultural materials–like church bulletins, small publications, and menus–saved and made accessible for future use. Through her archived correspondence and published excerpts in contemporaneous journals and magazines, we see that Buttolph was engaged in multiple preservation
projects. The notice of a letter to the editor of the *World’s Fair Bulletin*, concludes, “Miss Buttolph has also sent several numbers of the *World’s Fair Bulletin* to Edinburgh University. She says: ‘This whole series is too valuable to be kept in private hands.’” The quality of an article on prisoner of war camps “induced [her] common sense to file the *Confederate Veteran* at once in the Astor Library Reading Room,” she wrote in a letter to the editor of that magazine, adding, “This course met the approval of the Library officials.” Buttolph collected, filed, and sent documents to a range of institutions, primarily the New York Public Library, but also the library of the nearby Cooper Union and institutions such as the British Library and the Library of Congress. The menus, then, are far from a single-minded obsessive project. While the menu collection was Buttolph’s largest undertaking, it was part of a wider practice of engaging with institutions to save documents that would carry meaning for future readers.

Buttolph’s practices can be traced by examining the Frank E. Buttolph Papers at the New York Public Library. Her papers include hundreds of letters she received during the time she curated
Menu from the Pennsylvania Railroad, November 1, 1917, from the Frank E. Buttolph Menu Collection, New York Public Library.

the menu collection. Buttolph created most the collection by requesting menus from people and organizations in writing. She wrote hundreds of letters to restaurants, rail and ship companies, social organizations, chambers of commerce, government agencies, printers and trade journal and newspaper editors. At her request, contributors regularly sent her menus, some for over a decade; others collected on her behalf; and a number

http://curatingmenus.org/articles/when-a-woman-collects-menus/
gave her leads for other stewards, establishments, or texts that might help her collection building. Even those who didn’t send menus would often write explanations for why they didn’t have any to contribute and would commit to doing so if they could in the future.

She seems to have moved categorically through various sources. For example, a correspondent suggested that though they didn’t have copies of their own menus, their printer might. Her response was not simply to write their printer, but to write a number of printers, seizing on this kind of business as a new source for the collection. While one can see her personal interests in some of the collecting directions (writing to many Sons of the Revolution chapters for event menus), she seems to mainly be interested in breadth and completeness (getting a copy of menus from all the routes of a particular railroad company). Handling the postage and mailing for her expansive letter-writing campaigns was one of the ways that the NYPL agreed to provide support for Buttolph’s volunteer efforts.

The ways that correspondents engaged with Buttolph—the information they provide her, the letters of hers that
they refer to—form a picture of a woman deeply invested in collecting, who also generously shared materials, writing, and thoughts. While her papers reveal that she could be aggressive about the condition of menus, non-responsive correspondents, and errors in print and bibliographic records, they reveal her interest in preserving and understanding cultural history (particularly, she seems to have an investment in American history). She pursued information and materials pertaining to the meaning of flags, the experience of war veterans and memorialization, and the lives of revolutionary-era Americans. And of course, she and her correspondents frequently discussed the cultural and historical value of having these menus collected.

In addition to letter writing, Buttolph placed ads in newspapers and trade magazines like the Hotel Gazette. She worked with members of the press to write about the collection regularly. Though she was not always pleased with the outcome of the articles, she received new menus from people who read about the project and continued to engage with the press through the early 1920s. This publicity work
simultaneously grew the collection and
drew attention to its value.

Buttolph was also invested in curatorial
practices that focused on long-term
preservation. She was quite particular
about the condition of the menus she
received (unbroken and unsoiled) and
about the conditions they were kept in.
Early in the project, she complained to
the director of the library because the
menus were rubberbanded together,
which would leave marks on them. She
did a range of preservation work
“mounting on cards rare silk menus,
pen-printing others, smoothing out and
mending them which were crumpled ...
because every menu must be
absolutely perfect to be preserved.”
At the outset, she even suggested
having the menu collection kept closed
to the public for fifty years as a way of
preserving its content for future
historians.

By reading the evidence that Buttolph
left about her own work, we can begin
to see past the broad stereotypes
about collectors attached to her. The
fact that she volunteered at the NYPL
for more than twenty years to do this
work demonstrates her personal
commitment. However, her approach
to that work and the menu collection in
Mounted silk menu from dinner to Count Ferdinand De Lesseps held by Citizens of New York at Delmonico Hotel, 1880, from the Frank E. Buttolph Menu Collection, New York Public Library.

particular, was not a personal fetish but an investment in institutional preservation and future research.

**Buttolph in the Library**

We can deepen our understanding of Buttolph and her project beyond what she and her correspondents might have thought to note or mention explicitly by considering the library within which the menu collection was created and
curated. The twenty-year association between an unmarried middle-class teacher and the still-young New York Public Library is one of the most interesting and suggestive aspects of the menu collection’s and therefore the data set’s provenance. The milieu of the late nineteenth-century research library—and indeed specific features of the Astor Library and the larger NYPL organization into which it was subsumed—suggest how narratives that position the menu collection as exceptional (and its creator eccentric) are not reflecting simple absences of factual detail. Such narratives about Buttolph and her collection also reflect gendered ways of understanding library history.

The Astor Library is central to Buttolph’s story. Though the entity formally named “The New York Public Library, Astor, Lenox and Tilden Foundations” came into being as a merger of the Astor and Lenox Libraries and the Tilden Foundation in 1895, the research division of the NYPL, for which Buttolph volunteered, existed as separate physical locations at the site of the Astor and Lenox Libraries until the iconic main library building at 42nd Street and 5th Avenue was opened in 1911. Buttolph’s “career” at
NYPL was almost evenly divided between the Astor Library building in Lafayette Place and the “new” building further uptown. By association, the Astor Library underscores the research-focused nature of the menu collection and Buttolph’s serious intent in creating it.

The Astor Library opened in 1854 as a public reference library founded by private philanthropists and directed toward scholarly interests. The gift to establish the library came about through the lobbying of Joseph Green.
Cogswell, a scholar, educator, and bibliographer who had served as the college librarian at Harvard, traveled widely in Europe, founded a progressive school, edited significant scholarly journals, and tutored the children of New York elites. It was in these elite social circles Cogswell met and became friends with the wealthy merchant John Jacob Astor. So, while the donation was Astor’s (in a codicil to his will), the vision for the library, its collections, and even the design of its building were largely Cogswell’s.\textsuperscript{27}

In an early report to the library trustees, Cogswell expounded on his vision for a scholarly library: “There are but few general libraries in this country ... and here, in this great city especially, one was needed to supply before existing deficiencies: one that would enable the scientific enquirer to track the progress of knowledge and discovery to its last step.”\textsuperscript{28} The need for an endowed reference library for the serious pursuit of knowledge was deployed in arguments about the non-circulating nature of the Astor collections, about raising the minimum age to use the library to sixteen from fourteen, and about the need for an analytical catalogue (which became Cogswell’s own project until his
retirement). The result of this ethos was a well-known and well-respected library but perhaps not a well-loved one. A biographer of John Shaw Billings, the first director of the consolidated NYPL, diplomatically characterized the Astor Library at the time of the merger as having “concern only for the committed researcher.”

The scholarly bent of the Astor Library extended even to its architecture. Just as Billings would be involved in the plans for the NYPL building on 42nd Street, Cogswell had been heavily involved in the plans for the Astor building fifty years earlier. Housed in an early Victorian Italianate structure, the design of the Astor Library’s interior space was inspired by European and specifically German models. By the 1870s and 1880s, librarians would be pejoratively referring to this type of architecture as “alcoved book halls.” “These buildings have lofty rooms and a large open space surrounded with alcoves and galleries which are used for the storage of books,” wrote William F. Poole in a Library Journal article summarizing the most common features of such spaces. Poole was intensely critical of these designs from the perspective of the professional librarian—he accused boards of
trustees of new libraries of copying the worst features of existing buildings. Though, as historian Abigail Van Slyck points out, “while it is easy to imagine a donor relishing the comparison of his gift to one of the great European libraries of the past, the appeal of [this] library formula is more deep-seated than mere vanity.” Such designs Van Slyck argues “were particularly successful at articulating the family metaphor that sustained nineteenth-century philanthropy.” The combination of imposing public space, careful control of where visitors were allowed to move within the building, and domestic coziness (within the smaller alcoves), “library users were at once in a public institution and in the bosom of an extended family” (headed by the munificent donor). Partly in response to shortages of space and partly perhaps trading on this family metaphor that Van Slyck articulates, the staff of the Astor library selectively allowed visitors past the main charging desk to sit and work in the alcoves.31

These “alcove reader” privileges allowed researchers better access to sections of books related to their interests. Alcove readers were required to sign a logbook upon entry—giving their name and a few details of their
particular research project. We suspect Buttolph’s interest in genealogy may have brought her to the library. By the late nineteenth century, there was new and growing interest in genealogical research (a subject that had been treated with some suspicion in the early life of the United States) and specifically, leading up to and after the centenary, in finding and preservation information about Revolutionary War-era ancestors. From a brief inquiry published in the *New England Historical and Genealogical Register*, the first American genealogical periodical, we know that Buttolph lived not far from the Astor Library and that she was seeking information about what appears to be a personal ancestor who served in the Revolutionary War. Though alcove reader privileges were phased out in 1896, not long after Billings assumed Directorship of the library (to make room for more book stacks), it is perhaps not far-fetched to imagine Buttolph spending several years prior visiting the Astor Library, signing in as an alcove reader, and pursuing her genealogical research in a section devoted to American history. When Buttolph was then granted an alcove to work on her collecting project after the end of alcove privileges for ordinary patrons, this communicated in
real physical terms the location of her work within the larger NYPL.

Buttolph’s attitude to her collection may have reflected the tenor of the library around her in another significant way. A New York Times article from 1906 mentions Buttolph’s “ceaseless personal supervision” of her menu collection, implying, a few paragraphs later, that this is (in the author’s jocular view a somewhat comical) defensive vigilance on the part of an “unostentatious, literary-looking lady whose bugaboo is a possible spot upon one of her precious menus.”34 This journalist’s characterization plays neatly into stereotypes about collectors, especially women collectors, but we might consider how this depiction ascribes to Buttolph as personality quirks what were probably common attitudes to the library workers around her. One of the complaints that librarians made against the design of library buildings with numerous alcoves was that this design made it impossible for employees to monitor readers from a central service desk.35 In the 1870s, vandalism of books at the Astor was problematic enough to rate a story in The New York Sun and to cause conflict among the library staff over responsibility for the problems.36
Readers were expelled from the Astor for defacing and damaging books—which while still precious and possibly expensive could probably be replaced. Buttolph’s collection was composed of unique ephemera—as the trials she encountered in her collecting made plain. In this light, Buttolph’s relation to her collection is less the greedy protectiveness of a collector and more a reflection of professional attention to preservation of the collection materials for future use.

As we’ve discussed above, in the glimpses we have of Buttolph describing her own work, much seems to hinge on her desire to “preserve” the things she collected, including of course, the menus. An awareness of the preservation function of libraries, perhaps forcefully imbued by the milieu of a scholarly reference collection, seems to have been part of what distinguished collecting at and for the library. To return to one version of the collection’s origin story, what leaps out is the way that Buttolph herself saw her efforts as part of a larger library project and how she thought of the library as an agency that she could use for the ends of cultural preservation. “I stopped in the Columbia Restaurant for lunch and thought it might be
interesting to file a bill of fare at the library. A week later the thought occurred [sic], why not preserve others?,” she wrote. Buttolph’s use of the phrase “file ... at the library” is intriguing—why choose that particular verb to describe her project? With its managerial connotations, “filing” suggests the perspective of someone already embedded in the departmental work of the library. A patron bestowing a gift might “deposit” or “donate” a collection, but they would probably not “file” it.

Whatever the reason, the fact that she referred more than once in correspondence to filing things at the library suggests that she felt empowered to do so. At the time Buttolph began her official project, Billings was engaged in enrolling many new types of materials into the collections of the New York Public Library. Once again, Buttolph’s menu collecting was not “a tiresome freak” as the editor of the Literary Collector would have it, but one more contribution to the diverse collection building the library was undertaking at the turn of the twentieth century. During the decades when it was only the Astor Library, the funds available from the original donor’s bequest were
insufficient to meet all of the institutions expenses, and it was difficult to convince other patrons to support the library when someone else’s name was above the door. Thus, when Billings took over the combined libraries, the collections, while significant, were rapidly falling behind other peer institutions.

Billings directly supervised book purchases but also emphasized aggressive collecting of pamphlets, periodicals, reports and public documents. In 1899, Billings initiated divisions within the Reference Department for Jewish and Slavonic collections in recognition of New York City’s changing demographics. As in most libraries, gifts were an important source of collections. Due in part to Billings’ “catholic approach to collecting ... [and] his interest in ephemera,” half of the acquisitions made during the library’s first decade were gifts, of which many were pamphlets and documents. In the same year the Jewish and Slavonic collections were begun, the NYPL accepted a collection of three hundred prints with the intention of building a department devoted to collecting graphic materials. As Dain, the foremost historian of the NYPL
observes, “the development of these special departments was comparatively uncommon at the time and represented Billings’ characteristically pragmatic approach.” The significance of this for our understanding of Buttolph is not only that her extensive collecting was in sync with the purpose of the broader library around her but also that there was no fixed organization of library materials and departments for her menu collection to fail to fit. In its contemporary context Billings’ acceptance of Buttolph’s project amid the other activities of the growing library would not seem unusual.

(Miss)ing Histories and the Work of Frank E. Buttolph

In 1921, Frank E. Buttolph’s alcove was emptied out, many of its materials crated and sent off to her, and she was basically dismissed from the library (as a volunteer, she could not be fired). The reason for this dismissal is not completely clear, but letters from Buttolph and from library staff, patrons, and the director suggest that she had come into increasing personal conflict with the people working around her. The NYPL file also includes charges
that, in retrospect, seem trumped up to accelerate her dismissal. She was charged for instance with stealing books and with using the alcove for non-library collecting. Buttolph denied these charges, proclaiming repeatedly that her work was always for the library and for future historians. In 1924, she died of pneumonia at Bellevue Hospital.

This seems like a sad ending. The surviving records and letters about the end of her association with the NYPL might seem to corroborate all the narratives of eccentricity and mania that circulated and have continued to circulate about Buttolph. She did not like the whistling of the pages who fetched books from the stacks; she chided people about how they were caring for books; she called the police on children playing in the park next to the library. However, the fact that Buttolph could lose her temper and complained about the practices of her fellow humans does not undermine her assiduous and important work. It does not make her collecting less professional or her collection less a research collection (just as it would not were she a sometimes unpleasant man). In some ways, the kinds of complaints she registers, throughout the two decades she is at the NYPL, to
her coworkers, to editors, to collection contributors, reflect her investment in order and precision and, often, in getting the historical record right and having full and broad preservation of menus as evidence for future scholars.

The point of our work to expose the stereotypes in these narratives, to restore some context to them, and to retell them — is not to make Buttolph a saint, but to understand how her work and her life have been framed in order to better understand the meaning we can make of her collection.

The ways Buttolph’s actions have been recorded and interpreted reflect larger cultural constructions and interpretations of the data made from her collection must take seriously the experience of women in shaping libraries and the role of gender as a category through which those experiences have been recorded and narrated. In the NYPL’s collection and in most other places (including until recently Wikipedia) Buttolph has always been referred to as “Miss Buttolph” or “Miss Frank E. Buttolph.” While her name—Frank short for Frances—isn’t actually that unusual for a woman of her generation, the use of a feminized term of address seems to have become integral to how people understood her.
Even while she was living, Buttolph’s correspondences include various foibles concerning her name and gender. She often chooses to sign her own letters (Miss) Frank E. Buttolph—acknowledging her gender and marital status, but doing so parenthetically. Another frequent misassumption is that Frank E. Buttolph was her husband. If so, she would properly be Mrs. Frank E. Buttolph. More significantly though, this framing—as a philanthropic wife rather than a working woman who invested her own time and resources making a research collection—is no better.

As Clare Beck traces in *The New Woman as Librarian: The Career of Adelaide Hasse*, outspoken, competent women did not fare well with their male supervisors in libraries in the early twentieth century. Beck follows the story of Hasse, who worked at the Los Angeles Public Library, the Government Printing Office, and the NYPL. Hasse created new systems of collecting and organizing public documents and is now recognized for her significant and innovative contributions to librarianship. However, she was accused of stealing documents from the GPO, and her position at the NYPL ended in a host of accusations and
complaints about her being a “difficult woman, selfish, bad-tempered, and unreasonable.” The parallels between the type of work that Buttolph and Hasse were doing (extensive use of correspondence, building and organizing large collections of primary source materials) and the kinds of cases that were built against them (around their personalities and the use of institutional resources) suggest that despite the difference in their professional positions, they share a cultural position of being hyper-competent female research librarians in environments that were hostile to their intellectual contributions.

Buttolph was and is, like all humans, not fully legible. However, her collection is far from the idiosyncratic hobby project of an eccentric woman. Instead it reflects the rise of research collecting in libraries during the late nineteenth and early twentieth century and with that an investment is broad, publicly-facing, long-term use and historical value. The ways she obtained and maintained her collection—her aggressive collecting and preservation strategies—are part and parcel with the creation of research collections, particularly of primary source
materials, in libraries of the early twentieth century.

**Acknowledgements**

Thanks to the New York Public Library Manuscripts and Archives Division, especially to Susan Waide, and to the New York Public Library *What’s On the Menu?* project members, especially Dave Riordan. Thanks to Brian Vivier and Jen Guiliano for reading earlier versions of this piece. We also appreciate the support of Martha Brogan. Finally, thanks to CLIR for creating the occasion for us to meet and for championing scholarly work in libraries.

**Please cite as**


**Notes**

http://curatingmenus.org/articles/when-a-woman-collects-menus/


7. Lepore 133.


13. Quoted in Ryley.


19. “Literature of Eating; Collection of Menus in the New York Public Library,” *Evening Star* (Washington, DC), June 20, 1903,
20. “Most Interesting Array of Menus in the World.”


26. Frank E. Buttolph, letter to Mr. Galliard, July 19, 1921, Edwin Hatfield Anderson Records,

27. Dain 3-10.


29. Carleton Chapman, Order out of Chaos: John Shaw Billings and America’s Coming of Age (Boston: Boston Medical Library in the Francis A. Countway Library of Medicine, 1994), 290.


34. “Most Interesting Array of Menus”

35. Van Slyck 6.

36. Dain #.

37. Dain 130.

38. Dain 122.

39. Dain 122.


42. Beck 149.
refuse the old means of measurement.
rely instead on the thrumming
wilderness of self. listen.

out west, donika kelly

In the Digital Humanities, it is common to weigh the research potential of collections as data by evaluating their representativeness. That is to say, we ask to what extent the data have the capacity to characterize a person, an event, a period, or an experience. Where the data exhibit significant informational paucity, indeterminate values, inordinate biasing, or limited scope it is common to cast them aside in pursuit of something held to be more representative. Alternatively, a move is made to systematically qualify data absence as a means of shoring up grounds for a redefined notion of representativeness to stand upon. Both responses generally fail to engage with data absence as a feature rather than a bug to be quashed. How might data driven scholarship be conducted in a manner that centers data absence?

I turned to Twitter with a question in this vein and did my best to document the generous response below.

Amalia S. Levi shared Lauren Klein’s *The Image of Absence: Archival Silence, Data Visualization, and James Hemings*. With this piece, Lauren (1) demonstrates how Digital Humanities techniques can be used to address archival silence (2) and frames challenges that an archive of slavery poses for the Digital Humanities.

Scott Weingart referred to absence as, “more a creative wellspring than a lacuna”, and shared a concise presentation on *Fidelity at Scale*. Scott raised the notion of a workshop or conference focused on productive explorations of archival absence at scale. I am all in for that. I’d guess that others would be to.

Jer Thorp shared Mimi Onuoha’s *On Missing Datasets*. With this project, Onuoha calls attention to, “blank spots that exist in spaces that are otherwise data-saturated.” Onuoha is careful to emphasize that “missing” should be understood as, “a lack and an ought: [a
space where something does not exist, but should.” Onuoha goes on to introduce social factors that can be used to understand why data might not be accounted for. It should come as no surprise that these factors are arrayed against the most vulnerable among us. As a speculative exercise in seeing what isn’t there, Onuoha provides “an incomplete list of missing datasets.”

Ryan Dunn introduced the work of Andy Kirk. Andy shared a presentation on *The Design of Nothing*. In this presentation Andy provides a principle driven exploration of how to produce data visualizations where, “what is not happening is just as relevant as what is”. It is a kaleidoscopic presentation.


Jacqueline Wernimont shared Morris Eaves *The Editorial Void: Notes toward a Study of Oblivion*. Eaves presents a lengthy discourse on how to work with a history that, “even in the best imaginable circumstances, comes down edited by the harsh disciplines of purposeful and accidental forces.”

Mitchell Whitelaw flagged the notion of “intrinsic / representative incompleteness” raised by Tim Sherratt’s *Seams and edges: Dreams of aggregation, access & discovery in a broken world*. In this piece, Sherratt works against the perception of a seamless experience on the web in order to help us see how various people, data, and systems come together to constitute it – an exercise in finding the seams. With recognition of seamful experience in place, Sherratt asks, “What might happen if instead of seeing the seams and edges of our information landscape as speed bumps in the onward march of progress we recognized their fragility, and celebrated them as sites of collaboration, negotiation and repair?”
Clemens Neudecker shared his and Alastair Dunning’s *Representation and Absence in Digital Resources: The Case of Europeana Newspapers*. Clemens and Alastair present challenges they encountered addressing absence in the context a large scale historic newspaper digitization effort. How might a user interface for newspapers visually represent absence? How might the user ascertain the representativeness of a subset of digitized newspapers relative to all known holdings – digitized and not digitized?

Cole Crawford and Karl Grossner both reminded me of the amazing work of Stanford’s Center for Spatial and Textual Analysis (CESTA). CESTA tools like *Breve* and *Palladio* help users identify absence in data. This interaction also offered occasion to give a shoutout to Karl’s work with representation of indeterminate temporal data.

Anna Neatrour indicated that the question of absence would be interesting to explore in the context of Native American collections at her institution, where named people are typically members of the Bureau of Indian Affairs or Missionaries. Guha Shankar suggested that field notes from this period are limited and proposed a strategy of collaborating with descendants of source communities. Kim Christen noted that, “Absence is assumed, but may also be willful not seeing that is at play”.

Collectively, these examples evidence promising directions in treating data absence as an integral feature rather than a bug to be quashed.

In Donika Kelly’s words they, “refuse the old means of measurement.”

May the refusal flourish.