Data Modeling in a Digital Humanities Context

Modeling in the Humanities

Despite persistent ambivalence about the concept of “data” in humanities research, there is a long and rich tradition of gathering and modeling information as part of humanities research practice. From the perspective of the digital humanities, that tradition now appears in retrospect like important prehistory for an understanding of data modeling. And that prehistory is significant not only because it shows how integral such activities have been to humanities research, but also because it reminds us of the distinctive complexities and challenges humanities data poses. While the terms “data” and “modeling” may be new, many of the activities and intellectual frameworks they entail are familiar and deep-rooted. In a general sense, we understand intuitively that specific theoretical approaches rely on concepts and terms that divide the universe of ideas in specific ways. For instance, literary periodization constitutes a model of history in which spans of time are associated with distinct stylistic patterns and, indirectly, with cultural, economic, and historical phenomena that are presumed to influence those patterns and their evolution. The literary-historical approach is in itself a kind of general model, within whose terms more specific models could be framed and debated (for instance, concerning whether and how one might distinguish the medieval and Renaissance periods, and where the boundary falls in different national traditions). And we might reject the literary-historical way of modeling culture altogether, in favor of a model that disregards periodization, or that is uninterested in historical change, or that denies the existence of “literature” as a category. Debates about method are ultimately debates about our models.

In a more specific sense, our models represent the shaping choices we make in representing and analysing the materials we study. As Michael Sperberg-McQueen put it in his keynote to the 2012 workshop on Knowledge Organization and Data Modeling, “modeling is a way to make explicit our assumptions about the nature of a text/artefact,” and this statement is importantly agnostic with respect to medium. Although the digital medium has brought these choices and representational systems into heightened visibility, they have been at the heart of scholarship since the beginning. A classic example is the critical apparatus in a scholarly edition, a form of knowledge management which might be said to originate with humanism itself. As pure content, the critical apparatus is simply an account of the variations among the witnesses to a particular text, which could be communicated through a footnote or a prose essay. As information, however, the critical apparatus has taken its current structured shape through two closely related processes. The first of these is the formalization of the information it contains, placing it under regulation so that all of the components are verifiably present: the lemma or base reading, the variant readings and their sources, and so forth. The second, and closely related, is the development of standard notation systems which enable that formalized information to be processed efficiently and consistently. The use of punctuation, standardized abbreviations, and other notational conventions to group, delimit, and document each variant makes it possible for a reader to process this information quickly and systematically and to perceive patterns—in effect, to do with the human mind what we now consider the hallmark outcome of good data modeling in digital systems. Digital scholarly

editions emerged so early in the history of humanities computing in part because they were able to build on a clear existing model deriving from a long-standing tradition of practice.

The more recent history of data modeling builds on this trajectory. It draws on the insights generated by informal models (such as the difference between types of variant readings), which offer a descriptive language and an underlying set of ideas but not at a level of precision that would support the creation of a formal model. It realizes the informational potential represented by existing formalizable models, such as the critical apparatus, or the structure of a dictionary entry, or the organization of a financial ledger, which possess all of the qualities requisite for formalization: a clearly defined set of informational items with clearly defined relationships. Research on data modeling has sought to express this information in ways that support computational reasoning, as a formal model: one that rests on a logical or mathematical basis, whose definitions are expressed using some formal constraint notation (such as a schema), such that the information being modeled can be processed and analysed with reference to the model.

This chapter will explore the significance of this shift for our research methods, for the tools of scholarship, and for our understanding of the relationship between our models and our theories of the world. We’ll first consider in more detail what the digital turn has done for modeling approaches in the humanities and digital humanities. Then we will discuss the kinds of intellectual traction formal modeling can provide for researchers—a point that is picked up more fully in the next chapter, and in Michael Sperberg-McQueen’s contribution to this volume—and the complex relationship between those formal models and the tools through which we express and work with them. Next we will consider the relationship between our models and the intellectual scenarios they seek to represent, the relationship between models and the tools we use to manipulate and process humanities data, the tension between models and data, and the forms of critical engagement we must exercise in using digital models in a humanities context. We’ll conclude this chapter with some proposals for a research and pedagogical agenda in the domain of data modeling in digital humanities.

1) The Digital Turn: Modeling in Digital Humanities

It is often assumed that the affordances of the digital medium have brought into being new ways of thinking about data and new kinds of questions that were not previously thinkable. But in fact historical examples reveal a long tradition of attempts to analyse and represent data, often representing great ingenuity in the face of limitations in the medium of print. A regularly cited example is the attempt by Teena Rochfort Smith in 1883 to present a four-column edition of Hamlet in which the Folio and the first and second Quartos are shown in parallel, together with a conflated text, with complex typography through which the reader can apprehend the specific passages that differ between versions. Isabel Meirelles’ Design for Information (2013) offers numerous samples of complex visualizations representing analysis by hand of mortality data, historical imports and exports, agricultural production, and attendance at the Paris Universal Exhibition. The members of the New Shakspeare [sic] Society in the 1870s developed notation systems for displaying metrical patterns in poetry, aimed at supporting a large-

---

scale analysis of prosody to assist in dating Shakespeare’s plays. And concordances were a common form of humanities research data (and one of the earliest forms of digital humanities data) until they gave way to widespread use of dynamic searching.

Although more or less formal information models can be found in a variety of humanities contexts, there are some environments in which their operation is particularly visible to humanities scholarship. One of these is (naturally enough) in the domain of information science, where it impinges on humanities research practice: in the controlled vocabularies and information systems of the research library. Reference works such as dictionaries, bibliographies, concordances, and catalogues represent another long tradition of strongly modeled information. Still another is to be found in certain kinds of paratexts: title pages, colophons, footnotes, indexes, tables of contents, running heads, and other systematic apparatus through which publishers frame the cultural intelligibility of the text. These are particularly interesting examples since some of these formal systems exist not as an aid to the reader but as an artifact of the work processes of publication itself: for instance, the printed signatures that assist in the ordering and assembly of the book, or the verbal formulae associated with the license to publish (“Cum privilegio” and equivalents) which are a standard component of title pages during periods when this kind of oversight was in place.

With this long history in mind, what does data modeling mean in a digital humanities context? The landscape is characterized by complexity and challenge. We inherit from the humanistic tradition a set of modeling practices and concepts that, while foundational, are often unsystematic, poorly understood by non-specialists, and invisible through their very familiarity. Complicating this relationship is the fact that, as Scott Weingart observes, “humanists care more about the differences than the regularities”\(^3\): even in the domains where formalisms and “regularities” are well-established, we are inclined to treat exceptions and variations as the phenomena of greatest interest. Furthermore, humanistic data is strongly layered: the artefacts modeled in digital humanities are created with a purpose by identifiable agents and they have a history which is part of their identity, and then undergo further processes of curation whose intentions and methods need to be kept visible. Museum and cultural heritage institutions have developed ontologies—notably the CIDOC Conceptual Reference Model (CRM)—in which concepts like provenance and purpose are explicitly represented. Our models thus in many cases need to represent not only the history of the artefact itself, but also the history of the ways in which they have been described and contextualized. Alongside this humanistic legacy we also inherit from the history of digital technology a deep, thoroughly elaborated understanding of data modeling that has found a home in some specific domains of the digital humanities: notably in areas including markup languages, network analysis, ontologies, and game studies. These are all spaces in which an understanding of the models themselves, and a critical and theoretical attention to their design consequences, has been central to (and tightly coupled with) the use of models in practical research.

That kind of tight coupling and its attendant expertise are now, ironically, being made scarcer by the interfaces and tools that have popularized the digital humanities to a new generation of scholars. Where early humanities computing engaged intimately with its data models—through the development of standards like the TEI Guidelines, tools like TUSTEP, resources like the British National Corpus—the great rise of digital humanities in the 21st century coincides, not coincidentally, with forms of digital

scholarship in which contact with the model is at a remove and in which the technical expertise necessary to uncover and intervene in the modeling that animates our digital systems is increasingly rare. Our goal in this volume is to bring data modeling back into visibility and to demonstrate its centrality to all forms of digital scholarship. In order to do this, we need to re-examine our modeling activities—those that are already familiar to us and those that arise in the digital medium—in light of the more rigorous conceptual framework afforded by traditions of formal data modeling arising in fields like formal logic and mathematics, whose foundational relevance is suggested in the following chapter on essentials of data modeling. Bringing these domains to bear may open up opportunities for greater formalism, greater clarity of expression, and a clearer understanding of the edge domains where formal modeling is not possible. We can also benefit from a detailed examination of specific modeling practices as they apply to specific kinds of information and specific forms of analysis: text, geospatial information, temporal information, visual information. And lastly, we need to understand the social, intellectual, and political contexts in which data modeling takes place: the circumstances that shape our data standards, the operations of constraint systems such as schemas and encoding languages, the role that ideology and discipline play in our modeling strategies. The chapters that follow explore all of these things. As a starting point, the Gentle Introduction to Data Modeling provides a foundational discussion of essential terminology and concepts, starting with the concept of the “model” and the distinctive intellectual work it performs. Subsequent chapters explore specific types of modeling and specific conceptual areas, and explore some of the politics and open research questions attendant on this work.

2) Gaining Traction from Models

The conceptual shift that digital humanities brings to the humanities is nowhere more visible or consequential than in the opportunity to formalize and exploit information models. As we have seen, humanities scholarship already makes extensive use of structured information: the digital medium adds several important dimensions to this structuring. First, in the digital medium it is possible to create a formal specification of the rules governing a given type of data: a model of the data. This model can be used as a kind of template for the data set, in abstract terms: it tells us the set of constraints within which the data operates. It can also be used to test whether a given set of data obeys these constraints. As Wendell Piez has shown (Piez 2001), this kind of test has its roots in the history of manufacturing, with the emergence of interchangeable parts; the use of gauges and testing mechanisms that could verify the specifications of parts manufactured independently made it possible to ensure that they would fit together properly when assembled. In the digital environment, this kind of validation and testing is valuable for similar reasons: it enables data to be created independently of specific tools and contexts of usage. In specialized cases it may also provide additional ways of learning about the data set, since it shows immediately the possible patterns the data can assume and also may reveal some of the assumptions underlying the data design. This kind of data modeling also facilitates collaboration by allowing communities to formalize standards that can be used as the basis for shared tools, and it serves a pedagogical role as well, by supporting systems that can prompt novice creators of data towards shared practice.

In addition to its practical value in relation to the data it governs, the formal data model (represented through a schema or other specification) becomes an object of study in itself. Scholars studying the history of critical editing can examine the apparatus of specific editions and learn inductively about how their editors thought about the critical apparatus and its components. If we wish to extend that study into the age of the digital edition, we can examine the editions themselves empirically, but we can also
examine their data models to learn what formal categories the editors sought to impose (for instance, a
distinction between orthographic and substantive variants). We can also compare the actual data to the
data model (using validation tools) to discover whether these categories were used in practice. In effect,
moving processes write our knowledge about the content and semantics of our data into that data in
formal terms, giving the data a kind of intelligence and self-awareness.

This “intelligence” in the data represents a crucial theoretical and practical asset. It’s now commonplace
to observe that computational processing offers advantages of speed and scale that can move formerly
challenging tasks—such as concordancing or rapid statistical analysis—into the realm of the trivial. We
can also observe that even within the digital realm, formal modeling creates opportunities for processing
and analysis that are not possible with data whose modeling is less explicit. A stream of undifferentiated
text—words and spaces—may express a novel, a collection of oral histories, a set of personnel records,
but however apparent those differences may be to a human reader, they are inaccessible to computation
until their structural model has been communicated in some way to the computer. That modeling might
be represented through explicit structures in the data: for instance, as records and fields in a database, or
through markup that writes concepts like “chapter” and “dialogue” and “speaker” and “date” into the
text in ways that we will explore in the next chapter. However, it might also be represented in an
algorithm that can read the unmarked data and infer the structure of an oral history interview from
notational cues. Either way, the structure we assign to or infer from the data forms the basis of everything
else we can do with it: small practical tasks we take for granted, such as sorting personnel records by
postcode and surname for a mailing, or complex research tasks such as analysing the gender dynamics of
dramatic dialogue in early American plays.

If we approach the task of modeling in a purely decontextualized way, as an intellectual problem, it is
tempting to let our ingenuity run away with us: working at ever-increasing levels of detail to create
models of ever-greater complexity. Strong data modeling creates intellectual opportunities but it also
represents several added costs. One obvious cost is the extra labor of creating the data: both the amount
of work, and the expertise it entails, will be greater the more detailed the modeling and the finer
distinctions being represented. Identifying all of the names in a document can be done nearly
automatically; distinguishing between the names of persons, places, organizations, and other entities
requires human intervention and subject expertise to achieve with accuracy; identifying the individual
parts of each name and associating the name with a specific named entity may require significant
research effort. A less obvious cost is the work of developing and maintaining the model itself. In the case
of an ontology like CIDOC-CRM, or a text encoding language like the Text Encoding Initiative
Guidelines, or a metadata standard like METS, the model results from years of work by large numbers of
experts, and the additional involvement of thousands of other contributors whose data and user
requirements have shaped the model. And finally, there is the least visible cost of all: the cost of the extra
complexity of documentation, training methods, and knowledge preservation over time that arises from
maintaining a more complex data set. The point here is not that these costs are prohibitive or unjustified,
but rather that good strategic planning involves balancing the costs and benefits, and focusing the effort
in areas that offer a clear advantage. Strong, costly modeling of data that will be treated as ephemeral is
as short-sighted as using poorly modeled data in a context where its limitations will impoverish research
opportunities for years to come.
3) Data Modeling in Tension with Modeling Systems

We have been discussing the modeling of humanities data in ways that emphasize the intellectual contours of the information itself, and these contours are of great importance because they represent what the data means to us, the way it operates in our own minds. Within that sphere, we can readily conceptualize a scholarly edition as a transcription of a document that has a certain structure, to which we have added annotations which comment on specific passages, together with an apparatus that supplies variant readings for individual words and phrases. But when we undertake to architect a digital artifact representing this information, we need to work within specific modeling systems that have their own structural properties. As we will see, relational databases emphasize repeatable structures that assume a fundamental similarity of records across the data set, while XML emphasizes a grammar-based structure that is more open-ended and documentary, but requires that the document be conceptualized as a tree. Each of these approaches might offer certain kinds of informational advantages: a database would naturally enforce the regular structure of the annotations and variant readings and would provide excellent tools for querying and analysing the results, but that record-based regularity would feel less natural as a way to represent the documentary details of the base text. XML would offer better provision for open-ended documentary structures (including the representation of mixed content) but its prohibition of overlapping elements might prove awkward in cases where several different annotations or bits of textual apparatus apply to overlapping passages of the base text.  

4 A non-XML-based markup language like COCOA or, more recently, LMNL would make it possible to freely annotate arbitrary segments of text without concern for overlap, but would sacrifice the informational value of the explicit containment and demarcation of data elements offered by databases and XML. Pragmatically speaking, each of these disadvantages can be overcome, albeit inelegantly. LMNL can be transformed into XML, restoring the explicitness of element boundaries; one can represent overlapping spans in XML using empty elements and pointers; one can represent mixed content in a database through various workarounds. And at some level, these models are isomorphic: given sufficient effort, each of these information formats can be converted into the others. But each approach has a certain natural logic that maps more elegantly onto some kinds of information than others.

These systems are evolving steadily, driven by dissatisfaction with their limitations and by a desire for more expressive models that are more congruent with current research methods and ways of understanding the objects being modeled. Even if a document is like a tree (or a file cabinet, or a network), or can be represented as if it were one of these things, in truth it is none of these and there are many situations where pretending otherwise becomes awkward and restrictive. Experimental systems like LMNL have come into being precisely as efforts to demonstrate the existence of alternative possibilities,

---

4 A fuller discussion of the challenge of overlapping hierarchies in XML is given in [Witt/Banski chapter reference]. This problem has played a significant role in the research history on digital humanities data modeling, serving as a focal point for discussions of the homologies between the structure of data representation and the structure of the objects or concepts being represented. Early treatments of this problem (such as the seminal “What is Text, Really?” by DeRose et al. [1990]) sought to show that the hierarchical nature of SGML markup reflected a deeper truth about the nature of text as an “ordered hierarchy of content objects,” and subsequent challenges to that position often retain the philosophical significance while reversing its polarity, arguing that SGML and related markup systems fail to represent text precisely because they uncritically adopt theories of text arising from the print tradition and its formalization of language as an information system; see for instance Jerome McGann, Radiant Textuality (Palgrave MacMillan, 2001). A more pragmatic tradition, represented by standards such as the TEI, treats the SGML/XML data hierarchy as a practical feature whose chief significance has to do with processing convenience, possessing tradeoffs that can be compared with those of other data representation standards.
and to explore the expanded potential they may offer. At the same time, our usage practices are also evolving and becoming more pragmatic, in part as a result of tools that mediate the data creation process and allow more complex processing that can overcome inelegances in the underlying modeling. In the days when every pointer had to be entered and checked by hand, creating a workaround for overlapping XML structures was cumbersome in ways that accentuated the mismatch between the encoding and the “real” information being represented. With the advent of tools that make help to automate this work (as well as tools that can process the resulting markup to yield useful results), the feeling of philosophical wrongness is a little milder.

When do these differences actually matter, and how? Are our data modeling systems simply tools to be understood and used pragmatically, or do they carry cultural significance that informs the data we create with them? In the early days of SGML its tree structure seemed deeply consequential, whether one understood it as a form of intellectual tyranny (with its suggestions of hierarchy and patriarchalism) or as a statement about the orderliness and intelligibility of documents. Thirty years later we remain aware of the tree primarily as a technical commitment that XML makes, and our decisions concerning whether or not to use XML are typically made not because of its rightness or wrongness with respect to the nature of texts, but because of its practical properties or inconveniences. But has familiarity inured us to philosophical questions that should be more vividly in our minds? And are such questions—if they do matter—issues of aesthetics or do they carry an ethical dimension as well? These questions are difficult to answer fully in the context of this discussion, although readers will find relevant discussion throughout the volume, but a few points are worth noting here by way of orientation. First, in many contexts it is possible to identify a principle of elegance or “good fit” in the modeling of data. This can be characterized by lack of redundancy in the data, use of data structures that correspond to our intuitions about the intellectual organization of the data (for instance, using a network structure with explicit linkages between nodes to represent a community of letter-writers), data elements whose level of abstraction and granularity matches that of our domain analysis, and an overall architecture in which making the information connections required for analysis is simple and direct rather than requiring elaborate or extremely indirect traversals. Data modeled in this way is likely to be easier to document, to explain, and to program for.

But further, it is worth bearing in mind that pragmatic considerations are often the strongest determinants of the shape of data, and these considerations often militate against elegance or a natural homology between the data modeling and the object being modeled. Database software was for a long time far ahead of XML software in its power and level of engineering sophistication, with the result that many projects used relational databases even for data that was arguably an extremely poor fit in modeling terms, purely for the sake of the advantage it offered in speed and ease of development. Conversely, a project whose practitioners are very familiar with XML might choose to represent what is essentially tabular data using an XML structure, because the effort and cost of creating a simple schema and display system were less than that of either acquiring a database program or developing a database application using a tool like MySQL. As a more extreme example, one often encounters projects using word-processing formats to capture data which is clearly tabular—and could benefit from representation as a spreadsheet or an XML document—only because the project’s editors are unfamiliar with other tools. The differences between these cases are instructive. In the first two scenarios, despite the arguably poor fit between data structure and modeling tool, the actual modeling of the data may be perfectly appropriate. As long as the identification of relevant data elements, the expression of their relationships,
and the constraints on data types (numbers, dates, controlled vocabularies, and so forth) are intelligent and strategically appropriate, the most important goals of the modeling work have been met. With some effort, the data from one system can be exported and imported into another to take advantage of practical differences in speed, convenience, or specialized processing options. However, in the case of the word-processor, the problem lies in the fact that important aspects of the modeling are simply not taking place: the tool does not possess the capacity to formally constrain or validate the data, so the “modeling” is only taking place in the transcriber’s mind. As a purely practical accommodation, creating data in this way is a reasonable first step (if there really are no alternatives) but only if there is an equally practical pathway to get the data into a more usable format. But as Michael Sperberg-McQueen observes in his contribution to this volume, this kind of non-formalized modeling also risks “vagueness and ambiguity”: non-formal models “make it easy for modelers to deceive themselves as to the completeness, explicitness, and correctness of a model.”

4) Modeling and the DH Tool Set
For the digital humanities, data modeling as an essential part of our evolving relationship with tools. Within this relationship, formal modeling becomes a necessity, imposed by the tools; we’re always making models (of some kind) whether we intend to do so or not. All digital tools operate upon some form of modeled data, whether or not they fully expose that model to us, so if nothing else, we are creating information that corresponds to our tool’s way of modeling information. When we create a spreadsheet using a spreadsheet program such as Excel or Google Sheets, the data we create is modeled within the program as a set of rows and columns which are understood as computable objects: if we want to compute the total of items in a column or row, the spreadsheet software is able to do this. If we use a word processing program to represent tabular data, we can get the same visual effect of rows and columns, but we can’t perform computation on them; the software does not model the data as a set of computable values but simply as a set of text segments to be positioned on the page.

These days, people are creating data all the time: making travel reservations, doing online banking, wearing mobile fitness devices, sending text messages, writing Amazon reviews. This modeling of this data has only a practical significance for most people: they expect the systems that use it to work, but they don’t have an intellectual stake in the way it is shaped. The decisions about how best to model this information are made by those who manage it: airlines, banks, app developers, and so forth. As data creators, academics have a different, more knowing relationship to their data: they create data that is going to be a persistent part of the research environment, and they act as both its creators, managers, and consumers. The stakes of the modeling decisions for research data are thus much higher, and to the extent that these decisions are mediated through tools, there is significant value—even a burden of responsibility—in understanding that mediation. And within the academy, the stakes for digital humanists are highest of all, since their research concerns not only the knowing and critical use of data models, media, and tools, but also their critical creation.

There are several different ways in which tools operate in relation to models. For one thing, they can control our creation and editing of data, by constraining our options and by providing feedback when we make an error or a correct choice. The model here is significant because it represents our level of control over the data and our intentions towards it. If our intentions align with those of the tool—for instance, using a drawing tool to create vector images—then the tool can assist us and serve as an ally. It can also serve to express the model and make it visible to us for inspection: for example, database tools often
provide a structural view of the tables, fields, and relationships so that we can see how they are organized. XML editing tools similarly offer various structural views of XML documents and schemas through which we can inspect their architecture. If our intentions run counter to those of the tool, or if we’re simply unaware of how the tool aligns with those intentions, then the outcome may be surprising or unsatisfying. For instance, if we use an XML-aware editor to create XML data, it can assist us by preventing us from inserting markup that is ill-formed or invalid. Conversely, if we use a word processor to author the textual notes for a critical edition, the word-processing software has no knowledge of the specific components of such notes (witnesses, lemmas, variant readings, and so forth): its model of the data is purely oriented towards page formatting. As a result, the “data” we create in the word processor can only model what lies within the conceptual model provided by that tool.

And furthermore, tools can also interact with our models through processes like interoperation and conversion: for example, when a tool like Microsoft Word ingests an HTML file, the ingestion process involves mapping the HTML data structures (elements for headings, paragraphs, hyperlinks, metadata, and so forth) onto Word’s own internal data modeling, which includes many but not all of the same concepts. The conversion may involve a down-mapping (e.g. the elision of the distinction between “list inside paragraph” and “list in between paragraphs”) or a shift in semantics (e.g. a mapping of “block quotation” onto “indented block of text”), or a simple loss of data. The more fully aware we are of these underlying models and the mapping logic different tools employ, the more control we can exercise over our data during its entire life cycle.

It is also useful to consider how different types of tools assist our modeling efforts, at different stages of our work. Some data creation tools invite or even require their users to do good modeling by foregrounding and enforcing the model. An XML editor can enforce the rules of well-formedness actively by providing tag completion, and passively by highlighting errors; an XML editor that is also schema-aware can prompt the users with valid element and attribute options. Similarly, database systems often have user interfaces for data entry that provide access to controlled value lists, enforce the use of appropriate data types, and report inappropriate or missing required values. But tools don’t need to expose their models directly to do a good job of supporting them. For instance, an XML editor can offer users a formatted view of their data in which the formatting of specific elements (color, italicization, indentation, and so forth) provides a visual reinforcement of their correct usage. Proofreaders might not know that the author’s name in a bibliographic record should be encoded with <author>, but they know that when the text is encoded correctly the author’s name shows up in green. Some word-processing programs attempt to reinforce consistent modeling of documents by offering users the ability to create styles that associate semantic categories (such as headings, quotations, lists, and the like) with formatting features (bold or italic text, indentation, bullets).

Tools for manipulating and analysing data also draw attention to our models, and here the model is significant because it represents the horizon of possibility for the data in relation to the tool. The model is in effect the data’s way (and the data creator’s way) of communicating with the tool about what the data “knows”: its potential to reward analysis. For example, the Juxta commons is designed so that it can accept both plain text and XML input: it “understands” XML to the extent that it can ingest it successfully. But it does not take any advantage of the XML data structures in performing its tokenization and collation analysis, so XML data is handled as a benign but not advantageous form of modeling (whereas a data format like RTF or JPEG would be entirely unintelligible to this system). In a more fully XML-aware version of this tool, XML could offer positive advantages: for instance, by allowing the tool to
ignore metadata, or enabling the user to identify specific components of the text for exclusion (such as page numbers or running heads) from the collation. To take another example: the Gephi network visualization tool is not only designed to read in predefined graph formats, but can also take in and make sense of any tabular data (in the sense that it can parse such data as a structure). However, in the latter it needs help in determining which columns in the data are intended as nodes, and which are attributes of nodes. Gephi makes no assumptions about the tabular data model, which means that the person providing the data has more options, and also more responsibility for the success of the analysis, than if the input format was more precisely and exclusively aimed at producing network graphs.

In a very similar way, the publication tools and interfaces through which data is published for general consumption reflect their own underlying models in the opportunities they offer, or fail to offer, to their users. Here we are considering the large-scale publications—thematic research collections, digital editions, electronic research databases, online journals, and the like—that frame our access to a large portion of the available research data. These publications typically expose their underlying data modeling very selectively and not very explicitly through components such as metadata fields in a search interface, sorting options in a results list, or the formatting of a reading display. For instance, if a digital edition gives us the ability to see a manuscript transcription with or without authorial revision, we can infer that revisions are something that is explicitly modeled in the data; if we can sort a bibliography by publisher, we can infer that the publisher is explicitly modeled in the data. As above, the model is significant here because it represents the upper horizon of opportunity for our work with the data, the limits on what we can expect the resource to do. However, that horizon as practically realized in the resource may be much lower: our access to that modeling is limited to the places where the publisher chooses to expose it. So we may be able to view authorial revisions in the reading interface, but the search interface might not give us the option of searching for words that appear only in those revisions.

Such limitations are sometimes the result of deliberate design choices, arising either from usability concerns (since most users want only basic searching, why clutter the interface with rarely used options?) or from a need to limit costs. But they can also arise from limitations in the tools themselves. Digital publishing systems (including XML databases, digital repository systems, and content management systems) are designed around a set of the most common user needs and expectations: the ability to search on basic bibliographic metadata (author, title, publication date), the ability to browse and read items, the ability to use metadata facets to identify similar items in the collection. These systems offer simple configuration options (akin to the “dashboard” in WordPress) which make it very easy to develop publications organized around these features. But to offer more idiosyncratic options—which might take advantage of something very specific in a given data set, such as authorial revisions—requires that one intervene in the workings of the tool at a much deeper level. For instance, many TEI-aware publishing systems (such as XTF or Philologic) automatically index a limited set of TEI metadata elements that are needed for basic searching and display. But if one wants to index other data elements—such as markup representing textual revisions, or dramatic dialogue—so that these can be used to nuance a search, some custom configuration of the tool may be required. Some tools (tending towards the more complex and expensive) anticipate this customization and provide straightforward mechanisms for accomplishing it, while others may permit it only as a species of hacking, in some cases so as to effectively rewrite the tool itself.

These are important critical considerations, with impact not only on how our data is published and used but also—in a self-reinforcing way—on the expectations users bring to these research tools and hence on
how they frame their research in response. But there are also strategic and practical considerations that affect our design of the relationship between tools and data. An oversimplified version of these would state that data should always be modeled in a manner that is completely independent of specific tools, and indeed of all considerations concerning specific tools that might process it. The motivations for this statement are partly based on experience: the early history of digital humanities was populated by horror stories about research data trapped in formats only readable by specific word processors, or only usable on specific pieces of hardware. They are also partly curatorial: whatever our own intended purposes for our data, we understand that it may have potential research value for others that we cannot foresee, and that potential is heightened when we design our data without tool dependencies. The creation of metamodels like SGML and the relational model was strongly motivated by a desire to pull the modeling of data away from consideration of specific tools: to enable the same data to be used in many different contexts, and to abstract the essential processing of data structures (for instance, XML parsing) away from more tool-specific actions such as formatting or editing. The philosophical urgency underlying this position (which originated in economic considerations) draws as well on a sense of the importance of open data in the research community. The use of browser-specific HTML tags as a competitive weapon between browser manufacturers during the 1990s illustrated clearly how poorly the entanglement of data with a specific tool serves the creators and users of that data, and how tool-specific data could risk sabotaging the potential of a public data resource like the then-emergent World Wide Web. Similarly, early HTML modeled textual information in a way that was strongly aimed at web browser software aimed at displaying “web pages” on full-size computer screens, but the dissemination of web-based data has broadened to include other devices and other uses of the data. HTML has had to evolve in a way that abstracts it away from specific tools and separates data modeling from presentation, making it a more powerful and flexible language. Tool agnosticism thus enforces a kind of imaginative discipline, asking us to model our data to be as pure as possible an expression of the information we care about.

That discipline is a first and necessary move in a modeling process that does take tools into account, but resists situating them in positions of power or intellectual primacy. If we have in mind a specific form of output for our data—a printed monograph, a network visualization, an interactive hypertextual narrative—our vision for its distinctive functions will necessarily be shaped by the genre of tool through which those functions will be realized. The specification we write for that output will include considerations of the data’s shape and specifications as well: for instance, the need to distinguish between different kinds of annotations so that they can be handled separately in the output interface (e.g. some as linked endnotes, some as marginal annotations that appear on mouse-over). A tool-dependent approach would be to have the data itself indicate how each note should behave in the output; a tool-agnostic approach would be to identify the underlying functional and semantic distinctions motivating the different behaviors (authorial vs. editorial notes, biographical notes vs. word glossing, and so forth) and build these distinctions into the modeling of the data. This approach provides a level of indirection between that modeling (reflecting durable scholarly categories) and the behaviors that are available or desired within a specific tool context, with the latter being controlled by a stylesheet or configuration file that can vary from tool to tool.

Other pragmatic factors also play a role. There are often multiple approaches we might take to modeling the same data, with equivalent intellectual results. For instance, the following are two acceptable ways of representing a personal name in TEI:

\begin{verbatim}
<persName>John Stuart Mill</persName>
<persName>Mill, John Stuart</persName>
\end{verbatim}
Given this equivalence, it would be reasonable to choose between them based on what our intended output software will handle most easily. For instance, if these names are going to be used to generate an author list (which will be alphabetized by surname) then the second option is preferable to the first, but if they are going to be used as the heading for a biographical annotation, the first might be better. If we need to support both options (let alone a more open-ended set of functions) a more explicit modeling would be best of all:

```xml
<persName>
  <forename>John</forename>
  <forename>Stuart</forename>
  <surname>Mill</surname>
</persName>
```

There may also be cases where it is practically beneficial, and theoretically harmless, to include information that will be useful for a specific tool, even though that information plays no role in the deeper scholarly modeling of the data. It is in this spirit that one might propagate identifiers to all `<div>` elements in a TEI document, anticipating that a particular publishing tool will rely on them in generating a table of contents. Similarly, it may be useful to have our data contain a record of the tools it expects. The only cost of this information is the added burden of maintaining and documenting it.

For the tool-agnostic purist, these pragmatic considerations might seem like a dangerous concession, the first step on the road to sacrificing the independence of our data. But in the encounter between our data and our tools there is an important heuristic element in play that should not be undervalued. Every digital practitioner knows that the quickest way to discover flaws in one’s data is to load it into a tool—any tool—preferably as part of a public demonstration. This is humorous lore but also expresses an important fact: our data may exist apart from tools, but it reaches its fullest realization through enactment, through an active exploration of the patterns and ideas it enables. In this sense “the tool” constitutes a realization (albeit perhaps only partial) of our intentions for the data, and a test of the coherence of those intentions. Mobilizing our data through a set of different tools—even “the wrong tools”—can reveal omissions and inconsistencies in the data, areas where our modeling is too sparse or too detailed, and cases where our modeling fails to support the analysis we are seeking. Particularly for the novice practitioner, good data modeling is something to be done iteratively, interrogating and refining the model through a dialogue with both the source material and the operational context of tools.

**The Eternal Struggle between Models and Data**

As we have seen, data modeling is an attempt to create abstract, formal representations of the real world, and any model is in some sense a compromise or accommodation between two very different things: the quiddity and contingency of the material universe, and the clarity and computability of a formal abstraction. How in practice we reach that accommodation, and what we sacrifice in reaching it, will depend significantly on the goals of the exercise and the work process we employ.

One approach to model creation is to first consider our existing theories. A scholar developing an XML schema to represent poetry might start from the position that she already knows something about this genre: poems include lines that possess qualities of meter and length; these lines may be grouped and those groupings may contain patterns of rhyme; larger groupings of lines may have formal names (like “sonnet” or “villanelle”). From these reflections the scholar can create a schema representing her theory
of poetry. As soon as she begins transcribing actual poems and encoding them using the schema, she will discover that her actual documents diverge from the model, and she will be forced into one of two positions: either refine and extend the schema (for instance, to accommodate omitted genres, or to permit additional structures such as stanza headings and annotations), or omit any data that cannot be accommodated within it. In other words, she must decide whether her theory is useful to her on its own (as a way of identifying the class of poems that match it), or whether her project is really to arrive at a theory of poetry (a schema) that reflects the true diversity of poetry in the world. That latter position may feel more sympathetic, but it is also dangerous, since it precipitates a process of refinement that can only conclude when every poem has been examined and every possible nuance accommodated—a process that is asymptotic at completion. And it does not really simplify anything, since she will still need to decide which texts are to be considered “poems” for purposes of testing the schema; in other words, she will need to admit at some point that she does have a theory of poetry that is a priori rather than purely derived from the actual world of documents.

Another approach would be to start from the bottom and work up: to decide what documents we are interested in considering, and observe their behavior, and derive a schema empirically from what we observe. In this approach, we might not even be starting with the idea of a “poem,” but might simply be examining all documents to see what structural phenomena they contain. As part of this process we might discover that some documents contain segments of text that are rhythmic in nature and contain rhyming patterns, sometimes grouped into regular structures, sometimes with descriptive or numeric headings. After encoding a certain number of these, we might find that our schema had more or less stabilized, and that the phenomenon of the “poem” was emerging as a distinct form of text with its own internal variation but also some general patterns. The schema would express both the general pattern and the potential for variation, and an examination of the encoded documents would show us how those variations are combined in practice and how widespread they are in the overall population of documents. We might at that point start to feel that we had a “theory of poetry” though we would remain prepared to adjust it based on further examples.

Of course in practice things are never so clear-cut, and modelers of both types being illustrated here will in fact probably work iteratively with both documents and schemas; the ready availability of existing schemas like the TEI means that one rarely begins the exploratory process from a position of complete agnosticism. What these imaginary narratives illustrate is not two different literal work flows but two different intellectual scenarios in which an abstraction either is refined and tested by, or emerges from, a sampling of the world. In this narration, these two processes appear to converge at the moment where the modeler decides to stop working: either because the theory is satisfactory or because she has run out of documents or because she is too exhausted to continue. But in fact the stopping point looks very different depending on whether one arrives there “from the top” or “from the bottom.” For a modeler who begins the process with an interest in the data, the stopping point can be understood as a practical decision: one has considered all of the data that is relevant to a particular research problem (e.g. a particular collection, the work of a particular author, etc.), or one has reached a point of diminishing returns where the appearance of new variations is very rare. For a modeler who starts from the top, on the other hand, the stopping point has more to do with how much complexity one is willing to tolerate in the model: in other words, what kind of theory one is trying to develop. From this perspective, it becomes less interesting to list exhaustively all possible variations (which may not reveal anything essential about the genre) than to discover the patterns that are taken as characteristic of the model rather than merely associated with it. This
kind of top-down modeling is aimed rather at developing a blueprint for poetry (which might be used to generate a new poem) than at developing an inventory of things that have been called poems.

The difference between the two approaches also becomes clear when the data takes its place in a workflow. During the prototyping process in a digital project, sample data and draft schemas may stand in for the eventual mature versions as developers and project analysts design things like search and display interfaces. If the schema is being designed from the top down—in other words, if a draft schema can be assumed to represent the essentials of a document genre, with only minor refinements expected from further document sampling—then design decisions can be made based on that draft with reasonable confidence that they will not be overturned by discoveries arising from further inspection of the document set. On the other hand, if the schema is being designed from the bottom up, with the possibility that new documents might offer an entirely new perspective on the genre, then the stakes of completing a survey of the document set would be much greater and the status of a draft schema much more tentative.

**Engaging our Models Critically**

As already noted, our relationship with digital tools is developing in the direction of higher and more seamless function. One challenge digital humanists face is that of keeping our models visible, both to ourselves and to others. For those who are simply pragmatic users of tools—driven purely by outcomes and unconcerned with how those outcomes are achieved—the disappearance or inaccessibility of the model may pose no problems and indeed offers many conveniences, much as the evolution of the automobile has reached the point where drivers are relieved of the responsibility for knowing how their engines work or being competent to explain or repair or improve them. However, when humanities scholars use or create digital resources that operate as research contributions in their field, the stakes are different. For one thing, the modeling decisions in play have a direct relevance to the research being undertaken. The question of whether to create a synthetic edition of Hamlet that draws on both quarto and folio versions, or to treat the quarto and folio versions of the play as separate entities that can be compared, or to treat each individual copy of the quarto as a distinct text (as the Shakespeare Quartos Archive does), will profoundly influence how the play can be studied and reflects the editor’s beliefs about the nature of texts and documents and the role of the editor in mediating them. The editorial leader of these projects needs a detailed understanding of how these approaches work in practice, arising from deep competence with editorial theory and practice and familiarity with the texts in question. And although the users of such materials don’t need to be able to intervene in these kinds of modeling decisions, they do need to understand their stakes enough to assess what sort of edition is appropriate for the research they wish to undertake.

For editors of Shakespeare, their research field is scholarly editing and its concerns include the handling of textual sources, their variations, and the historical and textual causes that account for them. Digital humanities is also a research field whose concerns include the representation systems through which research artifacts become usable data, and the tools through which we manipulate and think with that data. As we see in “A Gentle Introduction to Data Modeling,” these systems and tools represent an evolving body of knowledge with a challenging research agenda of its own. And where digital humanities engages specific humanities subject domains (such as art history, scholarly editing, biblical studies, and so forth) it also takes on additional, specific questions arising from those domains, such as how to model scholarly editions as digital information systems. The stakes for understanding our models are thus highest of all for digital humanists, who are responsible for understanding and explaining not
just how a given model applies in a given situation, but how modeling systems themselves are designed and make meaning. This domain of expertise also involves being able to critique the ways in which specific research materials are represented in specific models.

As humanists we are trained to see symbolic or ideological significance in representational structure. So while the aesthetics or “elegance” of our data models—which as described above is rooted in functional properties—may lead us to seek a deeper meaning in our data structures, the problem of how to understand the cultural meaning of such structures is a methodological question that still awaits a rigorously framed response. The work of critical code studies\(^5\) has demonstrated that program code and data can be read as a cultural text, but it is clear that the mapping of cultural structures onto data structures—for instance, reading the hierarchies of XML as representing a commitment to social hierarchy, or the rhizome of hypertext as a radical departure therefrom—does not work in any straightforward way. In particular, such readings need to provide an account of how conflicting ideologies might animate the same structural paradigm: for instance, reconciling the rhizomatic nature of the internet with its origins in industrial-military research.

The ability to use tools and modeling systems critically is of clear importance to humanists and digital humanists. But for the latter group, the domain of expertise being described here also involves the ability to intervene in this ecology by designing more expressive modeling systems, more effective tools, and a compelling pedagogy through which colleagues and new scholars can gain an expert purchase on these questions as well. The revision and improvement of our models is an especially crucial part of the digital humanities research agenda. As these examples illustrate, models are situational, perspectival, and strategic. As artifacts of scholarship, they are necessarily always adapting, and this adaptation is an important research topic for digital humanists. One of these is changes to the technologies of modeling: that is, to the underlying representational systems through which digital information is shaped (which we discuss in more detail in the Gentle Introduction to Data Modeling). For example, the emergence of linked open data over the past decade has been supported both by the establishment of effective standards for modeling and disseminating such data, and the growth of practices and social expectations supporting its creation. These developments have meant that expertly modeled data from specific domains can be accessed and combined flexibly, rather than remaining in isolation or striving for self-sufficiency. Another example is the emergence of microformats such as hash tags, which can be included in social media such as blog posts and online communication systems like Twitter: because they are created by users in the very act of communication, they represent a genuinely bottom-up approach to modeling, but at the same time they permit a degree of formalization for entities (such as people and events), concepts (such as the expressions of political support represented through hashtags like #blacklivesmatter), and strands of shared activity (such as the discussion of keywords and contested terms tracked through #curateteaching).

A Research and Teaching Agenda for Data Modeling in Digital Humanities

\(^{5}\) For instance, Mark Marino, “Critical Code Studies” (Electronic Book Review, 4 December 2006); Noah Wardrip-Fruin, Expressive Processing (MIT Press, 2009); Nick Montfort et al., 10 PRINT CHR$(205.5+RND(1)); : GOTO 10 (MIT Press, 2012).
This book is the first to our knowledge to treat the domain of digital humanities data modeling as a cohesive field, but that treatment is only possible because of substantial prior work in more specific domains. These include large-scale, sustained research efforts by organizations like the TEI and CIDOC, which have not only resulted in widely-used standards but have also produced a legacy of sophisticated methods and insights. They also include foundational theoretical research in domains like network analysis, databases, and logic, and the thoughtful exploration of narrative and textual structures in domains like hypertext and game studies. Also deeply relevant is the work done in human-computer interaction and interface design that explores how people work with information and with the tools that expose information for our use. The digital humanities draws on all of these strands and more in an attempt to understand how the shaping of data shapes our research questions, activities, and discourse.

Our goal with this volume is to bring data modeling into greater visibility, in greater detail, for a new generation of digital humanists. The contributions to this volume help to illuminate the landscape and show the work that has already been done, the modeling approaches that are in use, and the complexities we encounter in applying them. They also help mark out a research agenda for future work.

One crucial item in such a research agenda is a history of data modeling. Some of the chapters in this volume make a start on this project—notably Lou Burnard’s chapter on standards—and Willard McCarty’s earlier work on modeling offers background as well. Isabel Meirelles’ Design for Information provides numerous valuable historical examples of visualizations that illustrate the evolution of the modeling activities they represent. However, because of the rapid growth of the digital humanities as a field—and ironically, given that the field now has a history going back at least fifty years—there is generally too little awareness of the history of current debates and research questions, and of the separate disciplinary strands that contributed to the early shaping of the field. Historicizing digital humanities methods is a crucial priority, and a history of data modeling in the fields where its key terms and processes originate would be a significant contribution to this work.

Another area of urgent interest is the challenge of associating semantics with our models, and aligning the semantics of different models, which is becoming a more acute and practical need as linked open data becomes more central to digital humanities research and practice. A second area of importance, highlighted in Michael Sperberg-McQueen’s contribution to this volume, is the question of how we can usefully model scholarly debate as an informational layer with the same degree of formality (and hence the same tractability to formal processing) as the representation of research materials themselves. For example, as scholarly communities form around research aggregations such as digital archives, how can data modeling help us identify significant strands of discussion, points of disagreement, and divergences or convergences of practice?

A critical related domain, and one whose importance is already fully acknowledged, is the exploration of how to model uncertainty and ambiguity. Some of the major data representation systems (such as the TEI Guidelines) include provision for modeling uncertainty and ambiguity in specific domains, notably the representation of transcriptional and editorial uncertainty, but most modeling systems at their core require uncertainty to be represented in the content (e.g. through qualifying notations) rather than in the modeling itself. We also need further exploration of how to use information about uncertainty in the context of retrieval, display, and analysis.
A less obvious research area concerns the extension of our data modeling to include the modeling of processes. Some notable existing examples come from tools that support use-configurable work flows, such as SEASR\(^6\) (whose representation of a set of data analysis modules requires that their individual actions and interconnections be modeled, albeit at a high level of abstraction) or the now-obsolete Yahoo Pipes which operated on similar lines. Data curation protocols for digitization and data cleanup are another area of process modeling that is highly relevant (if not precisely central) to digital humanities. But in all of these cases, the level of modeling for the process itself is under-generalized, even if the interconnections between steps use standard protocols and formats, and in many cases it is under-formalized as well: while data curation protocols may be well documented, they are not represented with anything approaching a formal model. There may be much to be learned here from other disciplines, particularly the social sciences and natural sciences, which have had success in modeling processes: for example psychologists modeling the reading process, or physicists modeling processes like the fall of objects. As tools and work flows for complex data analysis become more central to humanities scholarship, it will become increasingly important to bring the underlying processes to visibility and to express them in ways that support formal comparison.

A further segment of the research agenda concerns tools. Of primary concern is the development of modeling and publishing tools that put more intelligent and complex modeling decisions into the hands of scholars—acting both as publishers and as consumers of data. But in addition, a fuller study of data modeling tools could yield important historiographic and theoretical insight. How have the tools we use for data modeling evolved over time? How do they express existing assumptions and practices? And how do they affect the way we approach the modeling process?

Among the most enduring and challenging sectors of the research domain is the problem of how to document the underlying meaning of data models in such a way that we can start to align different data models with one another and create crosswalks between them. Early approaches to interoperability have tended to focus on simplification and constraint, on the principle that only very strictly regulated data can be reliably converted into another modeling system. However, with the proliferation of scholarly data of higher complexity and nuance, the stakes are high for exploring methods of inter-model mapping that are more supple. Furthermore, when we broaden our view to consider the modeling not of individual objects but of systems of objects, we encounter a new set of challenges involving the management of variation and the balance between goals of system-wide consistency and representational accuracy. For example, in a publication like Scholarly Editing (the journal of the Association for Documentary Editing), the accommodation of editions that use somewhat different customizations of the TEI Guidelines entails the development of systems of stylesheets that can handle different encoding approaches. These challenges have significance for publishing workflows and for large-scale systems that aggregate digital resources, such as institutional repositories and digital publishing frameworks, and they raise additional research questions concerning methods for managing the differences between variant schemas and the varying data they govern. In the long term, these questions also concern data curation and the development of mechanisms for documenting and maintaining the integrity of data and systems over time. Ontologies may offer a possible avenue for mapping and documentation of this kind.

Spanning across all of the other items in this research agenda is a need for attention to the politics of data modeling: not only the ideological and cultural dimensions that inform all modeling activities (whether

\(^6\) http://www.seasr.org
acknowledged or not), but also the issues of power and information access that determine who participates in the creation of reference models and standards, and hence determine the shape of those models and standards.

And finally, we need general theory of data modeling, which treats modeling in the digital realm as a set of activities which share common features, which are embedded into cultural contexts in a similar way and which can be evaluated in similar terms even though the models (relational databases, XML) are markedly different and are based on different mathematical concepts.

Accompanying this research agenda is a complementary pedagogical agenda through which we can also continue to shape the digital humanities field. We need to teach literacy in the basic modeling systems and tools early on, ideally even before students reach the university. We need to emphasize the scholarly importance of modeling decisions even as we teach our students how to create and publish digital materials, whether those are games or research data or digital archives or creative works. The “how” of digital humanities needs to be accompanied by an equally compelling “why” that expresses the motivations and ideologies that animate these digital materials. And in a complementary way, we need to teach students to attend to the modeling decisions our tools are making for us, or preventing us from making, and teach them to be resourceful about keeping their data from being too closely entrapped by specific tools.

**Works Cited**


Further reading


