

## A View from IT

James Smithies <james\_dot\_smithies\_at\_canterbury\_dot\_ac\_dot\_nz>, University of Canterbury, New Zealand

### Abstract

As digital humanities projects grow in size and complexity university programs will need to adapt, balancing the needs of technological systems with the imperatives of the humanities tradition. While it makes sense to adapt the accumulated expertise of the commercial and government IT sectors, care needs to be taken to ensure any new approaches enhance rather than undermine the aims of the humanities generally. While digital humanists are uniquely positioned to help the humanities, care needs to be taken to ensure new project management and design techniques sourced from the IT world are applied critically and do not undermine the core aims of the discipline. If these caveats are kept in mind the IT world has a lot to offer digital humanists, however, especially in the field of Enterprise Architecture (EA), which aims to produce a holistic, high level view of technological systems with a view to understanding social and cultural as well as technological issues.

## A View from IT

### *Caveat emptor*

Unlike the often cozy cultures of small software shops and, at their best, university departments, the IT departments of government departments or traded companies tend towards factories. They have a tendency to be hierarchical, intensely structured and risk-averse: project managers will themselves be managed by project directors; teams of developers work off functional specifications written by systems analysts who in turn work off business requirements produced by business analysts; databases will be designed from class diagrams upwards by specialist teams and critiqued by database administrators experienced with enterprise-class persistence layers; release schedules will be managed by a cross-functional Change Advisory Board (CAB) composed of senior managers from across the enterprise. Although there is a growing willingness to import Agile methods into these environments, especially for smaller projects, it's frequently remarked that the method doesn't scale to large projects. The fact is, of course, that when projects start to grow beyond the \$4 or \$5 million mark, they become business entities in their own right. While they might function on a principle of planned obsolescence, as long as they are active the fiduciary and managerial functions need to be almost as robust as that of an independent company. Agile is fantastic for software development, but software development is only one component of a large project. Large projects don't only deliver software, they deliver "Change", leading the project to focus on "transition management" as much as software development. Complex teams develop within these environments, requiring in turn more scalable development, governance and maintenance strategies. And this is perhaps what interests me the most about enterprise IT: even the largest software development projects are dwarfed by the logistics of deploying them into human environments. Difficult as it may be because of massive disparities in scale and purpose, this editorial is my attempt to consider what IT might have to offer the digital humanities.

As impressive as some enterprise IT environments sound, few experienced professionals could provide you with an example of one (whether composed of 50 or 500 staff) that runs seamlessly and without a degree of stress: the industry is known as a challenging one. The obvious reason for this is that it is involved in engineering and maintaining complex systems, and complexity brings with it difficulty. Another reason is that the industry is young. Compare the 70 years since the development of Colossus [Copeland 2006] to the several thousand that have elapsed since the major engineering feats of antiquity. We are only just beginning to find processes that will genuinely assist our software engineering tasks. There are now a handful of broadly accepted methodologies that claim – and generally do – make the task of software engineering and maintenance easier (RUP, AGILE, PRINCE 2, ITIL), but few organisations have either the will or the means to deploy these approaches in a “purist” sense. For a variety of reasons, they will need to be tailored to specific circumstances and the most extreme financial or legal motivations, like Sarbanes Oxley or governmental archive standards, need to be present to justify the expense of full deployment.

2

This is a perspective that digital humanists might find useful. Although the enterprise IT world tends towards being over-engineered, and often seems to aspire to a kind of 21st-century Taylorist nirvana, it struggles with exactly the same kind of problems that the lone digital humanist does when confronted with a page of .php. The difference relates, of course, to scale and purpose: where digital humanists’ IT messes can usually be scrapped and restarted in a healthy spirit of experimentation and discovery, an IT mess in the enterprise world has to be maintained and further developed according to the dictates of the business, often over decades rather than years. This can result in spectacularly complex *mélanges* of workarounds and patches - a situation that even the most rigorous methods and large teams of people can only “manage”, never tame. Much of the over-engineering (in both technology and culture) is due to this fact rather than an innate desire on the part of the workers to function in that way. Experience has shown that unless rigorous development and operational methods are applied to building and maintaining enterprise software systems, life can get messy. Hence the dictum that “no-one ever got fired for buying IBM”: Chief Information Officers are quite happy to pay extra for the insurance of a trusted brand.

3

Two points bear comment here. Firstly, digital humanists might not only be the advance guard of a movement of the humanities into the digital world. They might also be, in effect if not intent, narrowing the gap between the commercial and scholarly worlds. For the first time since World War Two, a significant humanist movement has appeared that engages directly with methods and attitudes commonly held in the commercial and governmental worlds. Far from being a flight into cyberspace, the digital humanities might represent an emergent movement out of the ivory tower and back to an engagement with the “real world”. If this is the case, humanists might want to be concerned. Even if the intent isn’t there, the need to control increasingly complex technologies demands a degree of rationalization and control that runs counter to a great deal of our tradition. Secondly, if such a shift is occurring, it provides us with a potentially fruitful area of research. Mathew Kirschenbaum urges us to consider the entanglement of the digital humanities in the material systems we use [Kirschenbaum 2007]; there is equal reason to explore the entanglement of those material systems with the discourse and methodologies of the corporate and governmental worlds. It may not be a subject to everyone’s taste, but it offers historical grounding and solid intellectual leverage. We need only go so far as the role of DARPA in the birth of the internet, or worse, the role of IBM Hollerith machines in the holocaust,[Black 2001] to find evocative examples of our debt to dubious aspects of technological history. These are morally complex and decidedly humanistic topics, and as such they could become central to our discipline.

4

The entanglement of the digital humanities with the commercial and governmental worlds symbolised by DARPA and IBM is connected to the fact that digital systems (at both a hardware and software level) are engineered. And as Thomas Hughes points out, engineered systems are the products of highly complex social, cultural and political processes: they are embedded in the fabric of history.[Hughes 2004] Rosalind Williams likens their architecture to the layers in the Wachowski brothers’ film *The Matrix: Reloaded* (2003): as you descend to deeper levels, more is revealed, but it is only at the deepest engineering level that the truth really comes out.[Williams 2007] Similarly, while robust and satisfying exploration is possible at the uppermost levels (in the context of the digital humanities, perhaps “merely” analyzing textual content presented on the screen), a journey to deeper levels presents a more intellectually rounded experience. The question for the digital humanities is what role these broader cultural and historical contexts should play in the development of the discipline. My feeling is that the digital humanities are about more than code and

5

applications, and that students should learn about the culture and history of the tools they use as well. A slightly stronger position might suggest that digital humanists, because of their understanding of computing and digital culture, have a responsibility to pose these kinds of questions.

The question, then, is just what kinds of analysis digital humanists will bring to the broader humanist culture. One thing seems certain enough: while there are a few groups doing things on the fringes, <sup>[1]</sup> the community doesn't seem particularly interested in theoretical complexity or grand narratives. Most practitioners seem to be quite practical people who experience daily the laudatory effect of things breaking when they don't design them correctly or drop a semi-colon. It's unlikely that people like that are going to stand up and proclaim victory or special insight, because they harbor a sinking feeling that the code could break, a bug could be introduced. They might even have to reinstall the operating system and start again. It's not the kind of job that's conducive to proclamations of universal truth. What it is conducive to, however, is a sense of responsibility when it comes to questions of technology. Digital humanists understand technology, and coupled with their background in the humanist tradition, are extremely useful as analysts of our technology-obsessed age. Limited theorizing (a little is always healthy), practical daily experience with the subject-matter, the context offered by 2000+ years of history and culture. What more could be asked for? The enterprise world is certainly crying out for it. [Times 2011] While the humanist tradition has (almost) always eschewed riches in favor of a commitment to art and culture, we should remember that people are getting paid very good money to provide technical analysis to corporations and government agencies around the world. Demand is high for people who combine solid technical understanding with the ability to write cogent, focused, easily-digestible prose. Computer Science degrees have to pack so much into them that these skills often can't be taught, and many of the students might not be particularly interested anyway. This wasn't a problem when IT products were less embedded into businesses' core operations, but those days are long gone. Today, businesses need people who can answer those curly questions "What will happen if I deploy this \$3 million piece of kit into this particular human environment? What impact will it have? What will I need to do to decrease any financial, legal or cultural exposure? How can we design the technology to minimize these risks?" The answer is often "hire a consultant", but I see no reason why it can't be "read an article by this digital humanist I heard of" or "hire one of those smart graduates from XYZ University". It might take a while, but one day digital humanists' environments may well become complex enough that we'll have to devise new methodologies to make sense of that complexity, and that could make our students sought-after in the job market.

This has been the case, broadly, in enterprise IT. Going by anecdotes from colleagues (men as well as women) who have been coding since the 1980s, the transition from mainframe computing to networked PCs during the early 1990s introduced a degree of human complexity that demanded a fundamental change in thinking. The movement away from mainframes not only decentralized computing power to the desktop, it also embedded computer systems within complex human-driven organizations, destabilizing the safe mainframe model and requiring a fundamental shift from "computer design" to "solution design". By the late 1980s the complexity of distributed IT systems were beginning to engender new ways of thinking about how to design robust distributed computing systems, to the point where engineers were being forced to think on a continental and even inter-continental scale. The end result was a (still relatively new) discipline known as Enterprise Architecture. Part of my purpose in this editorial is to introduce this methodology to the digital humanities community in the hope it will seed new shoots of thought. I'm not sure how much of the practice will be taken up, but it might at least prompt some thoughts about the scale we limit ourselves to.

Unlike solution architecture, then, which concerns itself with the functional and non-functional specifics of a single system like a mainframe <sup>[2]</sup> and is therefore intuitively understood by anyone who has spent time building software or web applications, enterprise architecture presents a holistic view of *many related systems* and exists to ensure those related systems are at once compatible, extensible, and maintainable. Unlike solution architecture, which functions at the level of components and code, enterprise architecture functions at the level of standards definition, process and (importantly) purpose. It exists to ensure solution designers have enough knowledge about the broader technical ecology to design systems that are not only fit for purpose, but likely to function in an optimal manner both in their immediate deployment context and into the future.

Significantly, because of the vast scale and corresponding degrees of complexity that enterprise architects work with,

this is the point at which the known hits the unknown, leading their methodologies to take on some of the characteristics of the humanities themselves. It is a level of abstraction that should come naturally to most humanists, trained as we are in large-scale thinking. Just as it isn't possible to unravel the complexities surrounding the origins of World War One into a neat formula, so it isn't possible to reduce a description of a national health system to a tidy technical diagram that only includes discreet physical models. Logical models must also be used, and these need to be aligned to political purpose or organizational intent. At an enterprise level, technological systems function within a human environment that reaches beyond the archetypal "end user" and into the cultural and political spheres. Decisions about fitness for purpose are not only made through reference to basic technical architecture, but also more subtle issues like risk appetite, strategic direction, and relationships to desired or redundant capabilities. And hot topics include not only domain-specific issues like these, but general problems like the movement towards the cloud, adoption of open access models, social computing, mobile computing, security, and the implications of open data and the semantic web. While the practice is almost antithetical to the digital humanities' focus on do-it-yourself, organic growth and a decentralized 'networked' structure, [Scheinfeldt 2010] its ability to make sense of highly complex, heterogonous IT environments suggests it might offer perspectives we can use. The key here is that, unlike some of the more business and efficiency-oriented enterprise methodologies, enterprise architecture aims to provide a view of technological systems that assists with technical, commercial *and* human decision-making. It is the most "intellectual" of any of the IT disciplines (a fact which is often noted with disdain by solution architects, analysts, coders and testers faced with coal-face technical realities). In short, it grapples with large issues, and that sits quite well with a humanist sensibility. Indeed, and perhaps at the risk of stretching a point too far, if the technical nature of the digital humanities has signaled a rupture between the old and the new, (elements of) enterprise architecture might allow the closing of the circle.

John Zachman is widely considered to be the father of systems architecture. His paper "A Framework for Information Systems Architecture" appeared in *IBM Systems Journal* in 1987, in the context of an exponential rise in the scale and complexity of information systems, prompted in large part by the movement from centralized mainframes with dumb clients to the vast networks of individual machines I mentioned earlier. As he put it:

10

... since the technology permits "distributing" large amounts of computing facilities in small packages to remote locations, some kind of structure (or architecture) is imperative because decentralization without structure is chaos. Therefore, to keep the business from disintegrating, the concept of information systems architecture is becoming less an option and more a necessity for establishing some order and control in the investment of information systems resources. The cost involved and the success of the business depending increasingly on its information systems require a disciplined approach to the management of those systems. [Zachman 1987, 276]

System architecture, as my more experienced colleagues suggested, grew out of a need to manage the explosion in complexity attendant upon the movement towards the personal computers that dot our homes and workplaces today. At the true enterprise scale, of course, we are not only talking about swarms of networked PCs, but vast industrial computing systems that might process billions of transactions a minute and be composed of hundreds of separate sub-systems. While my own discipline of History always makes even the most complex computing systems seem manageable, there can be no doubt that at the macro level – where the systems touch human social, cultural and political structures – there is a need for methodological rigor.

11

Zachman spent a great deal of his article describing in detail how architects go about designing buildings, and the different views that are required in order for the architect, owner, contractor and sub-contractors to work together to a common goal, even when the task might be so complex that none of the people involved have a full understanding of what the others are doing. Extrapolating this to software engineering, Zachman proposed a framework that, if completed, would offer a complete, holistic view of any complex software system from the perspective of managers, analysts, architects, testers and developers. As long as the framework was complete, the business could be confident that the project would be a success, even if some teams were never fully aware how their piece fitted into the larger picture.

12

	Why	How	What	Who	Where	When
Contextual	Goal List	Process List	Material List	Organizational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organizational Unit & Role Rel. Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location details	Event Details

Figure 1. John Zachman, *The Zachman Framework*, [http://en.wikipedia.org/wiki/File:Zachman\\_Framework\\_Model.svg](http://en.wikipedia.org/wiki/File:Zachman_Framework_Model.svg). Accessed 4 August 2011.

In reality, and although still used today, few organizations have the resources or the will to design and document a system with the three dimensional rigor demanded by Zachman's framework. It fitted well with the period in the history of computing from the late 1980s to the 1990s when the IBM Rational Unified Process (RUP) and waterfall software development processes held sway, and where complexity was controlled by large teams, hierarchical management and strict division of labor: Fordism (with a strong element of Taylorism) applied to software development.<sup>[3]</sup> While Zachman's framework remains useful to any enterprise development project there has been a change in approach over the last decade to a more flexible focus on process and standards definition, business modeling, and strategic requirements analysis. More than being a rejection of Zachman's methods, the various new approaches reflect an increasing demarcation between solution (or perhaps "technical") architecture and enterprise (or, perhaps "organizational") architecture.

13

As with software development methodologies, there are several different enterprise architecture frameworks available, but the most widely accepted approach is that offered by The Open Group Architecture Framework (TOGAF). TOGAF was released in 1995, and was based on the "Technical Architecture Framework for Information Management" (TAFIM) developed by the US Department of Defense (DoD). The method aims to provide structure and process to the development and maintenance of large, highly complex information systems. The Group describes the framework as a "detailed method and a set of supporting tools - for developing an enterprise architecture" [TOGAF 2006]. It is free to use by any organization wanting to develop an enterprise architecture, and is run on a commercial open source basis, charging for publications and certification. Simply put, TOGAF provides a set of seven guidelines, models and procedures for developing an enterprise architecture that will help ensure the smooth functioning and interoperability of heterogeneous systems. Unlike the Zachman framework, which focuses on creating a complete set of plans for the development of homogeneous systems, TOGAF assumes a splintered, heterogeneous environment composed of many different systems, standards and methodologies. It is particularly focused on the development of capability models, frameworks and standards that will foster reuse and the successful implementation of open architectures, to assist with interoperability in complex environments and ensure organizations aren't locked into proprietary systems or constrained by single-vendor models. The opening pages of the framework point out that the approach is designed to be used either in whole or in part depending on the maturity of the organization in question and the task at hand; it offers a set of conceptual "components" as much as an end-to-end process:

14

The intention of dividing the TOGAF specification into these independent parts is to allow for different areas of specialization to be considered in detail and potentially addressed in isolation.

Although all parts work together as a whole, it is also feasible to select particular parts for adoption whilst excluding others. [TOGAF 2006]

TOGAF focuses on the development of an enterprise architecture function within an organization, and the development of that function over time by implementing architecture repositories and fostering EA as a specialization for suitable staff members. In practical terms, therefore, all that is required to start using TOGAF is for projects to refer to the framework, and develop and store enterprise architecture artifacts (such as their capability model, their development process, their governance model, their technology stack) in a repository for future reference.

15

The development of enterprise architecture methods in the digital humanities community may take some time, but it might produce benefits in terms of consistency of approach and therefore potential future extensibility. It would also offer softer benefits associated with the development of a new specialization within the community and recognition of the benefits of specialization generally. Aside from developing a healthy “ecosystem”, and the obvious financial benefits associated with the use of Enterprise Architecture (lower software development, implementation and maintenance costs, increased portability of applications, reduced complexity, reduced risk, increased financial transparency), there are three other compelling reasons for developing the specialization within the digital humanities: interoperability, critical analysis and political transparency.

16

The benefits of developing interoperable systems seem almost too obvious to mention, and to a large degree interoperability is enabled by the technological choices we have in front of us when creating new information systems (in a lot of cases it doesn't really matter whether a project decides to use Apache Tomcat, Zend or Glassfish on their web server, or Windows or Linux for their OS), but as the digital humanities scale to a global level and implement more sophisticated layers for data analysis and messaging this will become more important. This isn't to suggest that we are inevitably moving towards a situation where major digital humanities projects developed in isolation aren't going to be able to talk to each other, but it does mean that people developing those systems might be working without a full toolset, or might encounter integration issues down the track. At the very least I imagine it would be reassuring for solution designers of major digital humanities projects to be able to consult an enterprise repository to gain an insight into how similar projects approached similar design issues. In some cases I expect they would find work that has already been done for them, or perhaps a more elegant solution to a particular problem. A related benefit to interoperability, then, is reusability: a good enterprise repository should offer patterns that can be reworked and reused, resulting in turn in further enhancements in interoperability. The aim is to implement not only enterprise architecture, but a virtuous cycle of sharing, interoperability and integration.

17

At the same token, of course, the development of enterprise architecture for the digital humanities would foster analysis of the field at a “meta” level that is necessary if the field is to gain respect among more traditional disciplines. At the moment it is difficult to present a critical analysis of the digital humanities, because we aren't sure just what the digital humanities are. Indeed, the last year or two have been characterized in part by a flurry of activity focused on deciding just what the discipline is, with some progress but little agreement. <sup>[4]</sup> This lack of definition can be contrasted with my own discipline of History, where historiography, historical philosophy and method are sub-disciplines in their own right and have their own established journals. Centuries of scholarly research and debate informs historical practice. Attempts by digital humanists to define the discipline and prompt debate implicitly recognize this: an academic discipline without a tradition of self-analysis will never enter the scholarly mainstream. By working towards transparency, by defining the terrain, by questioning the relationships between the technology we use and the purposes it was designed to fulfill, enterprise architecture could well help us mature intellectually as well as technologically.

18

One of the key questions that this level of transparency and attendant perspicacity would raise relates to political transparency. The discipline is clearly driven in large part, for instance, by social networks like Twitter that are first and foremost commercial concerns. How deep is this relationship? Would it change if an open source alternative became available? Conversely, what are the implications of gating university resources like JSTOR, and do digital humanists have a role to play in opening them up? If we have a choice, do we choose proprietary or open source products? To what degree are those choices dictated by university IT departments already heavily invested in certain technologies and unwilling or unable to change? The development of enterprise architecture would lead digital humanists down a

19

path strewn with interesting questions, and in doing so offer a level of maturity the discipline has to strive towards if it is to be taken seriously.

In order to indicate what an artifact in a digital humanities enterprise architecture repository might look like, I've produced a tentative capabilities model. TOGAF recommends the development of models to map relationships between objects, processes and standards, and define the limits of the knowledge domain. An important point to recognize is that enterprise artifacts are never complete. The aim is not to create a finite body of knowledge, but begin a process of discovery that will enable more sophisticated levels of engagement and analysis in the future. The approach should be second-nature to humanists. I hope the model shows how useful (if not revelatory and certainly not unarguable) enterprise architecture methods can be. The main point to be gleaned from the capability model below is that the digital humanities function mainly within the upper tiers of the model: the Tools & Application, Content, and Theory / Method layers. This is understandable given the digital humanities are by nature content-oriented; the development of content is supported by a set of tools and applications, and methodologies which guide practice. While many areas of the model are either poorly defined or simply unexplored, there are clear indications that they will be 'filled in' as the discipline matures.

20

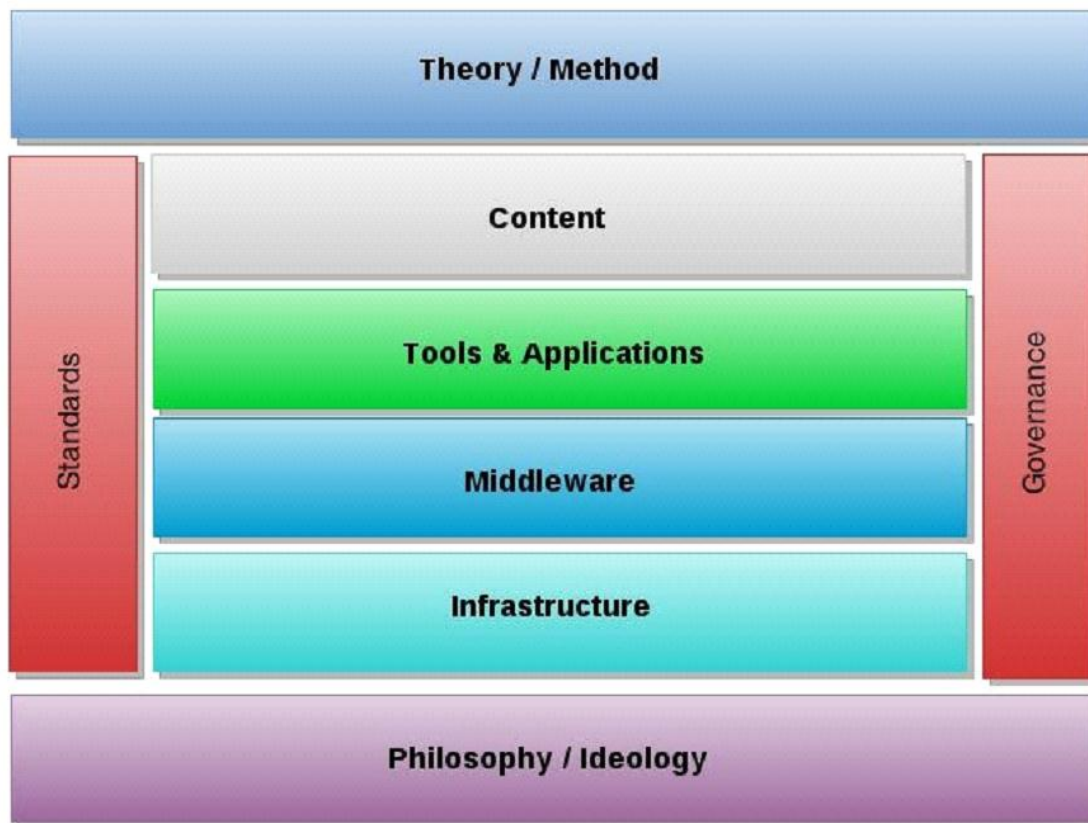


Figure 2.

## Theory / Method

This layer is self-explanatory, except insofar as it is anchored in the humanities proper, rather than information technology. Two representative texts are perhaps Dan Cohen and Roy Rosenzweig's *Digital History: A Guide to Gathering, Preserving, and Presenting the Past on the Web* [Cohen and Rosenzweig 2005] and Matthew Kirschenbaum's *Mechanisms: New Media and the Forensic Imagination* [Kirschenbaum 2007]. Both texts are firmly grounded in the analog humanities, and are effectively attempts to graft the digital humanities onto the main trunk of the tradition. Although a non-technical audience might find them intimidating, to most digital humanists they represent readable and penetrative overviews of the field. John Willinsky's book *The Access Principle* [Willinsky 2006] also fits into this layer (as well as the "Philosophy / Ideology" layer – see below), as do the essays on the Centre for History and New Media's essay list.<sup>[5]</sup> *Digital Humanities Quarterly* and journals like *Vectors* and the *Journal of New Media and*

21

*Culture* fill out the layer, as do practice-oriented sites like the Stanford Humanities Lab and machine-oriented sites like MIT's Media Lab. Indeed, the last 5 years have seen a marked increase in essays and websites that seek to define methods and theorize purpose for the field.

## Content

This is where the digital humanities were born, where I would hope the bulk of the activity still occurs, and where their core future lies. A decade or more ago, publications like Project Gutenberg, Paul Halsall's *Internet Modern History Sourcebook*, *Arts and Letters Daily*, the University at Buffalo's Electronic Poetry Centre, the University of Virginia's *The Valley of the Shadow* and the Center for History and New Media's *Liberty, Equality, Fraternity: Exploring the French Revolution*, provided clear evidence that good quality humanities work could be presented online, and that the internet offered a radical new channel for humanities output. Similarly, the arrival of Wikipedia may have provoked widespread panic about the quality of online content, but it is also the perfect example of how content-centric the digital humanities are. Above anything else, the new media requires content, just like the old media. This will only increase as the publishing industry transition more of their assets into the online world, new delivery platforms like the iPad proliferate, and universities come to realize the cost-efficiencies implicit in the online content model. When blogs and social media are included in the Content layer, it becomes clear that, for the digital humanities as for the rest of the online world, "content is king". There is little reason to launch into an extended discussion of this layer, because it is where most of us work on a daily basis.

22

## Tools and Applications

The Tools and Applications layer is another crowded space, but unlike the content layer, it is of little interest to mainstream humanities scholars. This is the layer where technical expertise takes over from content development, and in the digital humanities this normally amounts to development effort that aims to facilitate production and dissemination in the Content layer. It would be possible to argue for a separate Tools layer at this point in the structure, but I'm describing a single layer because tools and applications are bound by the Software Development Life Cycle: they are developed in code and inevitably reference each-other. For instance, at this layer we not only have operating systems like Ubuntu, Snow Leopard and Windows, but tools like Zotero and Monk and web applications like Omeka, not to mention the IDEs, code repositories and project management tools used to build them. Note that at this tier we also have the languages and frameworks used to write the applications: primarily .php and .css but also perhaps application technology stacks like Zend or web development frameworks like Ruby on Rails or Vaadin. Common content management systems like Drupal, Joomla, Moodle and Wordpress would also fit into this layer, as do developing web services like Google Maps and platforms like Hypercities. As the digital humanities scale both vertically (in size) and horizontally (into the "web of things") we may find Java being used more, and there is no reason to deny at least the possibility of DH-friendly languages appearing in the future, like Cobol and Fortran in the maths and sciences. This is a messy layer. It is essentially the engine-room of the digital humanities, churning out products that enable the growth of the discipline beyond 'mere' content creation. It actually needs a sub-model of its own. I suspect that most digital humanists would argue that, for the moment at least, this is the most crucial layer because this is the layer that will be filled completely by the large corporations if the digital humanities fail to expand into mainstream academia.

23

## Middleware

The Middleware layer could be the next area to be "built out" by digital humanists. I note this possibility because middleware will presumably provide a key layer in the growth of the semantic web and the web of things. Significant projects like SUDAMIH are already underway, focusing on the development of humanities-centric data management infrastructures. It could well be that the engineering of this layer (and "engineering" is the defining term to use for this layer, just as "development" is the defining term for the layer above it) is left to commercial organizations who provide "plug-and-play" middleware tools, but I suspect a lot of work will be done by computer scientists working in tandem with humanists, as with SUDAMIH and Project Bamboo. It isn't as though digital humanists are going to need specialist application servers or other similar products, but there are fascinating questions around the development of, say, an enterprise service bus (ESB) to facilitate data interchange across connected humanities infrastructures. With that in

24



place we would presumably be one step close to a global Service Oriented Architecture (SOA) for the discipline. Whether this is possible or even desirable at an international level is beyond me, and the term “middleware” does tend to escape precise definition, but it surely raises some interesting questions. Organizations like the Joint Information Systems Committee (JISC), Digital Research Infrastructure for the Arts and Humanities (DARIAH) and Coalition of Humanities and Arts Infrastructure Networks (CHAIN) are best positioned to investigate these questions, in combination with projects like SUDAMIH that are developing both systems and expertise. It will be interesting to see whether they view generic web middleware as adequate, or whether we need to put some effort into ensuring that the various international digital humanities systems can function at an inter-connected, global level. From their websites it appears they see some room for growth at this layer, even if it remains in fairly specialist, circumscribed areas. Defining the standards for the semantic web is one thing; ensuring our myriad legacy systems are interoperable is another thing altogether. The determining factor in the development of this layer will be whether the digital humanities' emerging Governance bodies (see below) decide that normalization is required to enhance interoperability across disparate infrastructures, or whether – as is perhaps more likely – a more loosely federated model is required, with an emphasis on clear articulation of standards and frameworks to prompt increasing alignment over the long-term.

## Standards

Our large public sector libraries have traditionally provided the most in terms of standards, stemming from their centuries old investment in classification and metadata, and their now decades-old move into the digital world. Like Governance, the Standards layer provides a crucial vertical structure to support the core Infrastructure, Middleware, Tools & Applications and Content layers. I have no empirical evidence, but my feeling is that Europe-based digital humanists seem more focused on this layer than US-based practitioners who tend to produce applications and content, and engage with social media, undoubtedly due to funding priorities. It's interesting to note that Project Bamboo have discussed not being “involved in creating standards, but rather in helping creators express their work”. Their notes point out the risk of “too much normalization” and “obstructing standards development” [Project Bamboo 2008]. This is understandable, especially in terms of TCP/IP protocols that can be left up to network specialists (and could be viewed as being part of the Middleware layer), but as I've noted above, digital humanists need to remain aware of the issues and perhaps identify areas where we can contribute in an efficient, focused manner. Following their discussion, Project Bamboo decided on an eminently sensible approach:

25

- articulating, documenting, and demonstrating benefits that would accrue to use of a specific standard
- adopting technical standards as necessary for internal Bamboo operation
- endorsing or recommending domain-specific technical standards (EAD, TEI, METS, VRA, etc.) which have methodological impact for scholars; also, basing higher levels of functionality or interoperation on the (optional) use of such standards
- communicating and raising awareness concerning the softest kinds of standards: e.g. “business practices” such as work flow, project management
- recommending best practices at all levels
- articulating the value of standards adoption for the community
- acting as a focal point for the humanities community's representation on technical standards bodies, to ensure that humanities perspectives are represented in the future development of these standards

[Project Bamboo 2008]

This approach leaves standards development up to the likes of those involved in initiatives like Dublin Core metadata and the Text Encoding Initiative and correctly places TCP/IP development outside their sphere of interest, while still providing an oversight and educative function for digital humanists.

26

## Governance

The digital humanities have no over-arching governance framework, which perhaps accounts for the slight lack of clarity amongst the discussants on Standards at Project Bamboo, who seemed to place TCP/IP middleware protocols in the same bucket as metadata standards, before discarding them again. Like Standards, the Governance layer provides a critical vertical structure to support the proper functioning of the core Infrastructure, Middleware, Tools & Applications and Content layers. This layer has only very recently started to be built out. Although not all directly related to the digital humanities, *Our Cultural Commonwealth: The Report of the American Council of Learned Societies Commission on Cyberinfrastructure for the Humanities and Social Sciences* [ACLS 2006], Peter Bradwell's *Edgeless University* [Bradwell 2009], and Cathy Davidson and David Goldberg's *The Future of Learning Institutions in a Digital Age* [Davidson and Goldberg 2009], all present overviews that could broadly inform future governance models: we are at the stage of initial analysis and exploration. Bodies such as DARIAH, CHAIN, JISC and Project Bamboo clearly have an important role to play in this layer too. The appearance of publications exploring the field, and the growth of organizations focused on governance and the facilitation of global interoperability is heartening. It suggests that the last layer in an enterprise model is beginning to be filled out and promises the level of co-ordination – or at least communication – necessary to the health of the broader system. Governance of the digital humanities is a thorny issue, however. It will not, and should not, be akin to the governance of an enterprise IT system: the field is too broad, the parties too disparate in their priorities, and the participants have a healthy disregard for processes and frameworks derived from the government and business worlds. Standard enterprise governance frameworks like ITIL and TOGAF are useful, but were designed for fundamentally different purposes than those of the digital humanities. At a philosophical level there will undoubtedly also be concerns about the intellectual lineage of such frameworks, based as they are on Tayloresque efficiency-maximising systems that are anathema to much of what the traditional humanities represent. But herein lies the problem: the digital humanities are fundamentally tied to the systems that enable the practice itself, and as the discipline grows so too will the complexities involved. It makes sense to pick models that have worked for other undertakings involving complex IT systems and apply them, critically, to our own. Perspectives like this should remind us that the health (and therefore viability) of the system as a whole relies on a balance between its component parts: the enterprise view isn't so much a normative view, as a guiding process. When a model loses its utility it should be discarded.

## Infrastructure

The infrastructure layer appears to hold the least interest for digital humanists, dealing as it does with the plumbing of the web, but it should not remain as neglected as it has been. Viewed from one perspective (the one typically found in the communications rooms of enterprise-scale IT departments), it isn't Microsoft, Apple or IBM that rule the IT world, after all, but companies like CISCO who provide the routers, switches and network infrastructure. In the future it may well involve companies who offer cloud platforms like Amazon AWS and Rackspace, both of whom are developing services of immense interest to digital humanists: on the one hand it appears it's becoming possible to outsource the IT department, on the other there are warnings from experts of the pitfalls of cloud services and the need for caution. Who is looking out for and providing advice to digital humanists on these topics?

This opaque world is understood completely by a relatively small number of specialists, but some things are clear enough: it is at this level that real control of the WWW resides, at the level of routing and packet switching. Indeed, this layer is so opaque that its main interest for digital humanists may well be the point where the discipline intersects with philosophy. (As Matthew Kirschenbaum has shown, there are equally beguiling problems to be found at the level of logic gates, hard-drives and the machine code that runs atop them.) The infrastructure layer includes a myriad of hardware and infrastructure: M9000 and P7 servers, blade racks and SATA drives, not to mention supercomputing facilities that measure performance in petaflops. Operating Systems could also be positioned in this layer, but note that these will often be IBM AIX systems, or other Unix-based ones like Ubuntu Linux, BSD, Solaris or Red Hat. They will not be made by Apple and may not necessarily be made by Microsoft. Whether this matters to digital humanists is moot in practical terms because they typically purchase services from either university or commercial providers and only ask that those services remain functional; as a discipline we are as much infrastructure agnostic as infrastructure ignorant. In political terms, though, digital humanists should surely be interested in the development of a safe and competitive ecosystem at this level as much as they are at the application level, with a healthy degree of variety and an emphasis on security.

Perhaps of more topical concern at this layer are the questions of data centers and virtualization: on the one hand we have giant data centers being built by the likes of Google and Facebook, who are both amassing an unparalleled degree of control over our personal data, and on the other we have widespread moves towards virtualization which looks likely to take us back to the days of thin clients and mainframes. The world's IT infrastructure is moving full circle back to the days where HAL9000 was conceived: massive, centralized computing centers accessed only by a monitor and keyboard. In our case, instead of time-sharing on a single operating system we may well be served our own copy of Windows, Apple or Ubuntu. The trend seems both economically and environmentally sound, but there is ample fodder for the novelists among us. The days of the Homebrew Computer Club, and the absolute level of control those days offered us, is coming to an end.

## Philosophy / Ideology

In terms of an enterprise model of the digital humanities, the weakest point could well be that of Philosophy. It is because of that that I've bracketed it here with Ideology, because it is my assertion that in the absence of a robust philosophical approach to the digital humanities we will only have ideology, which is not a pleasing prospect in the context of IT. Some of the potential topics for a philosophy of the digital humanities were touched on in my introduction to enterprise architecture and my discussion of the infrastructure layer, but there are many more topics that require investigation. Open Access and academia is one obvious area for further investigation. As I suggested above, I strongly support the open access and open source movements and believe the digital humanities should align to them. I also intuitively believe that the humanist tradition has a natural affinity towards them, but is this the case when it comes to university-based humanists? Academia has long been a bedfellow of the commercial publishing world, and is closely allied to both government and private industry. John Willinsky has given us good reason to accept the alignment between the academy and open access as a natural one, but all of us know academics who would vigorously disagree with him. Open source and open access principles are as likely to be championed and practiced by a computer science student or employee of a commercial IT company as an academic humanist schooled in the philosophies of John Stuart Mill. What does this say about the state of not only the digital humanities, but university-based humanities generally, and how would we gather evidence to test the assertion? Do we need a renewed commitment to open access and open source principles? Are we focusing enough on consciousness raising across the academy? Are universities so dependent on a commercial business model that we can't go open access without undermining the entire enterprise? Cultural and media studies have a lot to offer here, as do literary critics with a focus on technology and ideas.

30

Another interesting question is perhaps the history of ideas that binds classical logic with the logic of information systems, and the history of mathematics that details the development of algorithms and the methods used in object oriented programming. The history of technology and the humanities computing tradition have a lot to offer in this layer, putting the discipline into its proper context and reminding us of the engineered nature of not only the hardware, but the software that runs on it. Without this kind of research, digital humanists are apt to forget, or fail to adequately understand, the implications of working with two state machines. The issue runs deeper than the relationship between hardware logic gates and binary code: it bears upon a philosophical domain that sits somewhere between machine-level logic and frontier-issues like cybernetics and artificial intelligence. To locate this domain, we need to imagine a 'continuum of understanding' that takes us from two state machines, to the application of discrete mathematics and predicate logic to those machines in order to build creative work environments. This much is clear enough, but our understanding is severed here: as soon as we are presented with our creative environments we ignore the prior continuum and work as though our tool was as natural as parchment and pen. The continuum is then re-entered and we readily move to questions of cybernetics and artificial intelligence, quite forgetting that there is a huge gap in our humanist understanding of computers and the digital world at that earlier and far more fundamental interface between two-state logic and human creativity. Something seems to have been missed here. This is the precise point, after all, at which the (classically derived) logic that powers our machines enables the (classically derived) humanist logic that powers all that we create on them. It is, in sum, the event horizon of the digital humanities.

31

## Conclusion

A capabilities model such as that sketched here represents the "view from 40,000 feet". By its very nature it is both

32

general and liable to adaption and change. It is also, however, one of the tried and tested means by which large, complex information systems are conceptualized, governed and improved through enterprise architecture. As the digital humanities grow, it is no longer going to be enough to focus on code and content in the 'engine rooms' of the Tools & Applications and Content layers. There will be a need for some people to adopt a high-level overview of the undertaking as a whole so that they can create philosophies, practices and frameworks that will ensure the transparent functioning of the whole. Because as much as humanists may balk at a view of their world that seems to imply a growing leviathan, it is surely better to learn from the history of systems development and deploy (and adapt) methodologies that will help us understand the boundaries of our subject area, produce maximally efficient designs, and point us towards the more esoteric limits of our discipline.

## Acknowledgements

I would like to thank the reviewers of *Digital Humanities Quarterly* for their valuable comments on this editorial.

33

## Notes

[1] For example <http://www.vectorsjournal.org>.

[2] Codebase, data models, messaging structure, concurrency, availability and so on.

[3] Note that RUP is still widely used; it merely competes with a plethora of other approaches.

[4] See for instance the Stanford Humanities Lab, "The Digital Humanities Manifesto 2.0" (Stanford Humanities Lab, 2009); ThatCampParis, "Manifesto for the Digital Humanities", 2010; "philosophi.ca : Center Net 2010", <http://www.philosophi.ca/pmwiki.php/Main/CenterNet2010>; "The Landscape of Digital Humanities", <http://www.digitalhumanities.org/dhq/vol/4/1/000080/000080.html>; "Something Called Digital Humanities", <http://www.digitalhumanities.org/dhq/vol/2/1/000020/000020.html>; "How do you define Humanities Computing / Digital Humanities?", Taporwiki, [http://tapor.ualberta.ca/taporwiki/index.php/How\\_do\\_you\\_define\\_Humanities\\_Computing/\\_Digital\\_Humanities%3F](http://tapor.ualberta.ca/taporwiki/index.php/How_do_you_define_Humanities_Computing/_Digital_Humanities%3F); "Defining the Digital Humanities", Columbia Scholarly Communication Program, April 6, 2011, <http://scholcomm.columbia.edu/2011/02/10/defining-the-digital-humanities/>, 24 May 2011.

[5] See <http://chnm.gmu.edu/essays-on-history-new-media/essays/>.

## Works Cited

**ACLS 2006** Unsworth, John, et al. *Our Cultural Commonwealth: The Report of the American Council of Learned Societies Commission on Cyberinfrastructure for the Humanities and Social Sciences. The ACLS Report.* 2006. <http://www.acls.org/cyberinfrastructure/ourculturalcommonwealth.pdf>.

**Black 2001** Black, Edwin. *IBM and the Holocaust : the strategic alliance between Nazi Germany and America's most powerful corporation.* London: Little, Brown, 2001.

**Bradwell 2009** Bradwell, Peter. *Edgeless University: why higher education must embrace technology.* London: Demos, 2009.

**Cohen and Rosenzweig 2005** Cohen, Daniel, and Roy Rosenzweig. *Digital History: A Guide to Gathering, Preserving, and Presenting the Past on the Web.* Philadelphia: University of Philadelphia Press, 2005.

**Copeland 2006** Copeland, Jack B. *Colossus: the secrets of Bletchley Park's codebreaking computers.* New York: Oxford University Press, 2006.

**Davidson and Goldberg 2009** Davidson, Cathy, and David Goldberg. *The Future of Learning Institutions in a Digital Age.* Cambridge: MIT Press, 2009.

**Hughes 2004** Hughes, T.P. *Human-Built World: How to Think about Technology and Culture.* Chicago: University of Chicago Press, 2004.

**Kirschenbaum 2007** Kirschenbaum, Matthew G. *Mechanisms: New Media and the Forensic Imagination.* Cambridge: MIT Press, 2007.

**Project Bamboo 2008** Bamboo Workshop . *Standards and Specifications.* 2008.

<https://wiki.projectbamboo.org/display/BPUB/W2+-+Standards+and+Specifications>.

**Scheinfeldt 2010** Scheinfeldt, Tom. *Stuff Digital Humanists Like: Defining Digital Humanities by its Values*. *Found History*. 2010. <http://www.foundhistory.org/2010/12/02/stuff-digital-humanists-like/>.

**TOGAF 2006** The Open Group. *The Open Group Architecture Framework Version 8.1.1*. <http://www.opengroup.org/architecture/togaf8-doc/arch/>.

**Times 2011** Times Higher Education. "Google leads search for humanities PhD graduates". *Times Higher Education*. <http://www.timeshighereducation.co.uk/story.asp?sectioncode=26&c=1&storycode=416190>.

**Williams 2007** Williams, Rosalind H. "Opening the Big Box". *Technology and Culture* 48: 1 (2007), pp. 104-116.

**Willinsky 2006** Willinsky, John. *The Access Principle: The Case for Open Access to Research and Scholarship*. Cambridge, MA: MIT Press, 2006.

**Zachman 1987** Zachman, J.A. "A framework for information systems architecture". *IBM Systems Journal* 26: 3 (1987), pp. 276-292.



This work is licensed under a Creative Commons Attribution-NoDerivatives 4.0 International License.