

Is Software Production a Cultural Practice?
Position Paper for 2018 Working Group
Trinity College Dublin Centre for Digital Humanities
J. Edmond, M. Doran, G. Nugent-Folan

“There can be no doubt that the epistemologies and methodologies of computing and computer science are fundamentally influencing the shape of contemporary culture. ... But computer science is itself a part of culture” (Manovich: Cultural Software, 2011)

Introduction

Science and Technology Studies (STS) is a well-established branch of academic research, largely based in the empirical methodologies of the Social Sciences, and with its roots and inspiration in the technological revolutions of the Second World War. While STS continues to make significant contributions to the way we see and understand how technology is shaping the world around us, the fact remains that human capacity to develop digital technologies, products and platforms is overtaking society’s ability to assimilate them. The embedding of digital platforms and tools into everyday lives, with good and bad consequences for human identities, cultures, communications, agency, values and creativity, requires new methods and approaches to be developed to understand their impact. In particular, this embedding of digital technologies into the cultural layer of human interaction, proposes the question of how humanities approaches, as opposed to social science-based ones, might be more widely and effectively applied to questions of how digital technologies and platforms are driving culture and society today. In particular, we propose to test the idea through a framing conceit proposing that software development can be productively viewed as a culture: one with unique modes of communication, values, hierarchies, in- and out-groups, identities, power relationships and hierarchies, etc. From this perspective, the conflicts and biases of contemporary digital culture may be able to be recast in terms of cultural conflicts of the past, with a potentially productive outcome for the future.

Part 1: Why ‘Culture’?

Viewing software development as culture allows us to read the conflicts between society and software as intercultural relations: after all, software systems and the people that build them do display distinct languages, norms, shared understandings, biases, ethics etc. This holistic approach will hopefully not only complement approaches to the digital society growing out of in the social sciences, but also enable a paradigm allowing humanities methods and knowledge to take a leading place in the study of the digital in human cultures and lives.

There are many frameworks and definitions one might use to frame a study of culture. We hope to investigate the applicability of these in the future. For the purposes of developing this position paper, however, we fall back on to the canonical work of Sociologist Geert

Hofstede, whose interest in values in the workplace maps in a practical manner onto our questions of the work-based habits and cultures of software developers. Although not all of Hofstede's dimensions map precisely on what can be observed in software development teams, and although one does not want to risk falling back on stereotypes, reflecting on them in this context provides an interesting window for beginning to view software development as a work environment with distinct cultural overtones.

- POWER DISTANCE and INDIVIDUALISM VERSUS COLLECTIVISM both encourage reflection on the team-basis of so much software development work.
- MASCULINITY VERSUS FEMININITY seems an obvious point of interest, given the high profile critiques of the male domination of the tech industry.
- UNCERTAINTY AVOIDANCE seems rather baked into the practice of coding, which is very unforgiving of irregularities or uncertainties.
- LONG TERM ORIENTATION VERSUS SHORT TERM NORMATIVE ORIENTATION with technology changing so fast, historical or long term perspectives may inevitably be lost in coding circles.
- Finally, INDULGENCE VERSUS RESTRAINT may again be visible in the stereotype of the corporate office of the software age (Google, Apple) rather than the hardware age (IBM).

There would be other ways to approach the question of whether software development builds a culture around it, however. For example, one might look at components of culture, such as language (not just programming languages, but how one speaks about key facets of one's work, like data), values (an engineering mentality of solving problems and reducing complexity seems to prevail), heroic narratives, values, etc.. In the light of this last element, one might apply frameworks to question shared understanding among software developers such as Anderson's idea of imaginary communities. Do identities among software developers respond to similar ideas of in- and out-groups, and are such high-stakes placed upon them?

A final aspect that should be considered within the study of culture is that of acculturation. If values and habits can be identified in what is essentially a culture of immigrants (there may be habits of mind and thought that attract people to coding, but one would assume that few people are born into such an identity), then we may also usefully look at how young software developers are trained to absorb the values of their tribe, the incentives and penalties that exist, and indeed where the possible tops of the hierarchies may be found (for example, in training and degree programmes), as these could be entry points for possible investigation or even intervention.

Part 2: Why 'Software'?

In the preparation of this white paper, the correct term to describe the object of investigation has been one of the most contended questions. Where exactly does the origin of the digital culture lie? What is the most precise term we can use to describe it? Many alternatives were proposed: computing, software, algorithms, digital, hacker, developer...

Many of these are simply too broad: “computing” has a nice sense of the quantifiable, but many aspects of quantifiability are marginal to the phenomena we seek to describe below. “Computing” also has a strong implication of fundamental hardware, of chips and mainframes, which also seemed out of scope.

“Digital” is of course even broader, while “algorithm” seemed to miss key drivers of the current cultural shift (we need certain kinds of algorithm to make sense of big data, for example, but they are not necessarily the same things).

“Hacker” seemed to be getting closer, as it invites a focus on the human driver behind technological change, the people who write the codes and design the systems, but may not have the imagination to foresee the eventual uses and abuses of their creations. “Hacker,” however, carries an implication of a counterculture, of an intent to subvert. What is happening now seems, instead, to be much more banal, if not entirely evil. “Developer” is too broad, covering too large a semantic field to carry much meaning.

For the purposes of this white paper, therefore, we settle on the term of “software.” It is much easier to take a deterministic view of software than hardware, it is the rules that the machines follow. By and large, it is also the product of those very well-meaning developers, hackers and other engineers who seem to share a language and culture unique to their own in-group (a claim that will be discussed in the next section).

Part 3: Is there a ‘software culture’?

This is, of course, the key question to answer. Human identities are always fractured and multiple, weaving together influences and allegiances to families, countries, faiths, gender cohorts, etc. But the answer to this question should not take the form of a definitive ‘yes’ or ‘no,’ but rather of potential a toolkit that allows us to better understand how technology is not just shaping, but being shaped for society.

Part 4: Why the Humanities?

The social sciences have a long history of engaging with the societal impacts of information technology. To date, however, there has been comparatively little discourse within the humanities about how to best contribute to a greater understanding of the influence of the epistemics of computer science in defining modern culture. This is particularly problematic given that, as Gary Hall pointed out recently, “computer science is not all that well equipped to understand itself and its own founding object” (3).

The humanities approach to knowledge creation prizes itself on being source-led. While more experimental disciplines might view this as a flaw, representing a lack of rigour and basis for exact repeatability, it does mean that humanists are able to interpret complex objects and contexts in which no clear adjudication of right or wrong, statistically significant or not, can be made. Like hunter-gatherers, humanists collect in a more opportunistic, but in some ways richer manner, than the narrowed but more intensive field of the agricultural.

In many ways, computer science is seeing a convergence with humanities now in this way. Deep learning and other forms of AI are creating results that the system creators cannot necessarily predict, or trace back to their exact origins. A system in which interactions and relations are not entirely clear is one that, as a research object, approaches the complexity and messiness of a human one.

Humanities disciplines develop their understanding on the basis of the networks of artefacts, intentional and happenstance, that human beings lead behind. While this may tend to elide the individual dimension of human decision making, it does allow access to a deeper, nuanced evidence base regarding collective resonances and the relational aspects of identity formation at a very deep level. These are processes currently being transformed by digital technologies as well.

There is little mystery as to why a proposal like this might emerge from the digital humanities. The integration of data and computer science in humanities research represents a crossover of methodological and ideological approaches, indeed of epistemic stances, lending a unique perspective to people working at this interface. In addition, digital humanities has a significant 'hacker' culture of its own, lending an appreciation for the engineering imperative to make a system work, even in the face of conflicting, disorganised, messy data. But there are also voices from within the digital humanities community pushing both forward and backward to precisely the kind of enquiry this position paper proposes. Alan Liu, for example, has called for digital humanities to engage in a cultural critique of technology practices, and others like Gary Hall (quoted above) are recognising the utility the humanities may have in realigning the relationship between technology and society.

The traditions of humanities research may tend to the basic, rather than the applied. It will be one of the goals of this project to move beyond that, creating a space in which the humanities can move not only beyond disciplinary silos, but also into new audiences and translational dimensions for research.

Part 4: Possible Questions and Approaches

Some of the leading questions we have developed to frame this discussion are as follows:

Gendered, classed and racialised bias is relayed via everyday practices and exclusions. By adhering to these codes you perpetuate the performance of your gender, by repeating these simple coded behaviours every day. You do not have to exist on the "edges" or "fringes" of heteronormative society for these things to affect you. By and large, it's the performative actions of the majority that create and then perpetuate the gender roles that negatively impact the minority, or those that do not fit into an either/ or category. How do our digital practices evidence this?

Aristotle's Nichomachean Ethics argues that human happiness (eudaimonia) requires human agency (arete). Why therefore are we so concerned about machines either taking on human characteristics (such as creativity or imagination) or not having them (like ethics)? Why has complexity become coterminous with stress and frustration, or viewed as a risk to

be managed? How do we invest trust and delegate decision making to algorithms when we do not understand fully how they function (eg. black box, deep learning)? Do we believe that the decomposition of tasks into executable subroutines perhaps changes the nature of both the question and the answer (or do we care)?

To what extent has STS been cast as the carrier of a kind of 'emotional labour' in relationship to the technologies it explores?

It has been suggested that attention should be viewed as a human right and privacy as a public good. How do we draw lines between selves and others in data environments where we cannot see what (or whom) we expose? Should algorithms be regulated? Does the GDPR go far enough? And, what do we owe to others?

Software is generally designed to meet the needs of a client or a user. How can the needs of the wider society be taken into account? How can the tension between what a client wants to see and the requirements for decision making be resolved against a backdrop of software as a product which needs to be user friendly, to be competitive? How can the tendency of the digital to consolidate access, power and wealth be redistributed? As data inequality emerges as a major theme in data driven society, what are the consequences of this inequality in terms of technology, infrastructure and employment? Is there a potential to facilitate redistribution between the data rich and data poor and other kinds of economic inequality?

The relationship between narrative and data has received much attention from philosophers of science, being variously presented as antagonistic, antithetical or even symbiotic; existing in relationships that can pose ethical and epistemological challenges for the researcher or software engineer (Presner 2017, Rosenthal 2015, Hayles 2007, Manovich 2002). Less has been said about how this relationship is perceived among the computer science community. What does "narrative" mean within the context of the data-focussed computer sciences? How can one make data comprehensible if you define it a priori as patternless, objective input? How does the human need to make sense through storytelling become recast in an era that seems to fetishise data as having an almost god-like claim to truth?

With the growing push for Government regulation of AI, the "Driverless Dilemma" has emerged as a uniquely computational development on the well-established "trolley problem" wherein a driverless car must choose between the lesser of two evils. Rather than asking whether it is possible to design something that can make machines more humane, should we consider whether it is possible to design machines that make people more humane? Are we now tasked with moving towards a post-human algorithm, one that is free of the biases and prejudices (conscious or unconscious) that currently delimit human drives and human(e) choices?

Finally, with all of its jargon, block boxes and immateriality, how can we look critically at how we talk about digital technology? What lessons can we learn from this exercise?

Part 5: Joining the Discussion

In September 2018, the Trinity College Dublin Centre for Digital Humanities will convene an interdisciplinary working group of researchers from across the humanities, the social sciences and the computer sciences to address the question of how best the humanities can contribute to our understanding of the computational and the digital. Building on the findings of the EU-funded Knowledge Complexity Project (aka KPLEX, see <https://kplex-project.eu/>), we propose to invite scholars from across Europe and beyond to respond to our position paper, and join us for an interdisciplinary workshop investigating the current boundaries between STS and cultural studies, and how we might transcend them.

Objectives of the Working Group:

Through this method, we feel we should be able to:

- To elicit the humanities perspectives, including but not limited to perspectives from the study of culture, in discussions around the emergence and development of digital culture and society.
- To supplement, correct and refine the way in which the humanities are currently applied to the study of technology, in particular through insights from STS and computer science, but also with an eye toward how an applied and collaborative dimension for humanities researchers to lead interdisciplinary thought in this area might be developed.
- To create a network with a manifesto and shared purpose behind it.
- To disseminate and embed the findings from the KPLEX project, and inform implementation of the project's recommendations

We will also focus within the workshop on enriching and/or revising this original position paper, with a view toward releasing it as some form of publication (journal article, white paper, policy brief ...).

Participants Proposals:

Prospective participants are asked to submit a response to this white paper, informed by but not necessarily focussed on their current research, but positioned in terms of how this research might inform a wider dialogue. We want people to take risks, to learn from each other, and to expand their access to perspectives emerging in other disciplinary communities. This is not a conference to showcase empirical, technological, or indeed even humanistic research, but rather a sandbox, designed to spark new ideas and establish links between disciplines, or indeed move into a post-disciplinary space of shared expertise.

Expenses:

The Trinity Centre for Digital Humanities will offer reimbursement of return flights and accommodation up to a maximum of two nights following participation in the working group to all working group participants. The event will be fully catered.

For enquiries about any of the above, please contact: doranm1@tcd.ie (Michelle Doran).

***The organisers of the working group are grateful to acknowledge the financial ***
support of the Trinity Long Room Hub Research Incentive Scheme.