

Soft Materials: Assessing Architects' Roles as Ethical Producers of Digital Technology

MAYA PRZYBYLSKI

University of Waterloo

This paper exposes the ethical implications of custom computational components embedded in architectural projects. It argues that these components, identified as soft materials, constitute part of a project's material assembly and should be subjected to similar ethical considerations targeted at other, more physical, project aspects. This, in turn, necessitates an expansion of knowledge and methods and, to this end, we present work on building capacity for richer engagement with the socio-cultural dimensions of projects' computational components. We focus on data as a key component and present methods with which designers can develop a more sophisticated understanding of the data used in their work. These methods guide an exercise in unpacking a specific project's data components along dimensions, such as objectivity and accuracy, which have inextricable links to ethical concerns, such as inclusion and access.

If we don't address software itself, we are in danger of always dealing with effects rather than causes, the output that appears on a computer screen rather than the programs and social cultures that produce this output.

—Levi Manovich¹

INTRODUCTION

Architects are actively exploring the potential relationships, afforded by new computational capacities, between material, form, data, algorithms and project outcomes. With this, architects begin to play a dual role with respect to digital technology. On the one hand, they are consumers - using countless software packages to assist throughout the design process. On the other hand, architects have also become producers of digital technologies – developing custom computational processes that are embedded as agents in the design phases and beyond. This second role, the one in which architects directly produce new technology, is examined here in terms of its ethical dimensions and its accompanying requirements for new understanding and methods. How do we as architects begin to understand the ethical dimensions and relationships between immaterial project components, such as data and algorithms, and real-world project outcomes – be they material, formal, or socio-cultural, socio-political?

This paper begins to unpack the ethical implications bound up in the notion of architect as technology producer. It argues that computational components embedded in design work

constitute part of a project's material assembly and should be subjected to similar ethical considerations targeted at other project aspects. This argument is supported by articulating key offerings from the field of Software Studies which position software – its actual lines of code and data, not just its effects – as a material practice with both social and spatial outcomes. As such, software should be assessed as having agency with respect to ethical outcomes. This is a complex process that cannot be fully explored in the space of this paper; what follows is a preliminary work focused on building architects' capacity for richer engagement with the socio-cultural dimensions of their projects' computational components. Specifically, we focus on data as a key component of our embedded software, and present methods with which designers can develop a more sophisticated understanding of the data used in their work. These methods guide an exercise in unpacking a specific project's data components along dimensions, such as objectivity and accuracy, which have inextricable links to ethical concerns, such as inclusion and access.

ARCHITECTS AS PRODUCERS OF TECHNOLOGY

Architects now play a dual role with respect to digital technology. On the one hand, they maintain their well-established role as consumers of digital technology – using software packages for supporting computer aided drafting, building information management, digital modelling, rendering and other processes. Here software supports work through the design process and then typically ceases to be actively engaged once the project is complete. On the other hand, with the rise of the computationally-integrated work, architects are becoming producers of digital technologies. Using a complementary suite of opportunities ranging from extensions to familiar packages, such as RhinoScript and Grasshopper, to standalone design-oriented coding environments such as Processing, and numerous other less domain-specific languages and platforms, architects are integrating data-driven computational procedures into their design process and beyond.

Through the development of code, architects are implementing and deploying custom algorithms to do wide ranging work in their projects: supporting undertakings such as form generation, automated analysis, advanced visualizations, and interaction and responsiveness. The installation *Project Distortion II*, carried out by a team of researchers from the Centre for Information Technology and Architecture at the

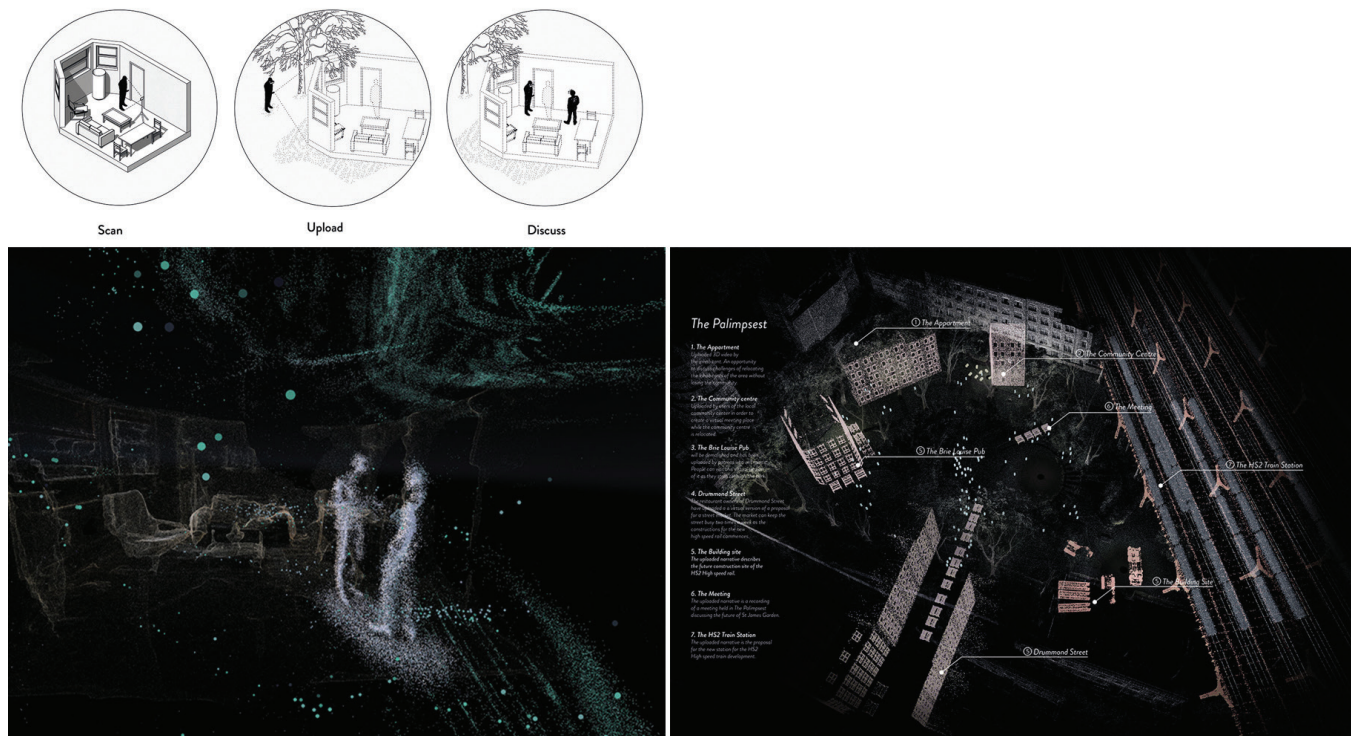


Figure 1: The Palimpsest. The project utilizes emerging digital technologies in virtual reality and 3D scanning to improve community engagement and participatory design in urban development projects. The Palimpsest allows communities to digitally archive neighborhoods, homes, and urban spaces that are important to them and that may be impacted by development projects. It also provides a platform for all stakeholders involved in urban development to collaborate by sharing stories, opinions, and proposals.⁵ Project & Image Credits: John Russell Beaumont, Haavard Tveito, and Takashi Torisu. www.johnrussellbeaumont.com/the-palimpsest

Royal Danish Academy of Fine Arts School of Architecture, investigates the potentials of acoustically heterogeneous spaces through formal and material articulation. With respect to form finding, the project used parametric methods to create a tunable digital model that responded to material and acoustic performance criteria allowing for various formal configurations to be tested via custom analytical tools during the design process.² Nicholas De Monchaux's *Local Code* project, an example of automated analysis, presents a collection of digitally tailored proposals for 3,659 vacant lots in US cities, all found through computational processes combining existing documents such as aerial photography and municipal zoning maps.³ *Sentient Chamber*, by Philip Beesley Architect, pairs information-inspired architecture with digital sensing and physical responsiveness, employing lightweight structures, sensors, and motorized actuators.⁴ A project combining human memories, city planning and virtual reality, *The Palimpsest* evokes the potentials of information-inspired visualization to create more inclusive planning practices, using emerging technology to directly connect communities, governments, and developers in conversation (Figure 1).⁵

Works like these leverage the increased spatialization of information communication technologies and their capacity,

together with design, to actively mediate urban life. Outputs by allied researchers and designers are incredibly varied, producing buildings, building components – such as façade systems, objects, installations and exhibitions, tools, and interfaces.⁶

In all cases, custom code is developed by writing sets of algorithmic sequences to process data in order to produce some output. In one set of scenarios, as in *Project Distortion II* and *Local Code*, code is active while the project is in development, and then its output, be it a related set of 3-dimensional geometries or 3,659 site diagrams, is disconnected from the computational apparatus when deployed in the physical world. In a second scenario set, as in *Sentient Chamber* and *The Palimpsest*, the code remains online and continues to actively mediate project outcomes and behaviors through time and space. While both of these scenarios are relevant in a discussion around *what actions, by architects as producers of technology, are necessary to define and contemplate with respect to ethics in the pursuit of a more just built environment*, this paper focuses on the latter case, labelled *software embedded design* (SED) – where the custom software actively contributes to the project's outcome in time and space, in the real-world.

As software is embedded in this way, a new set of responsibilities emerges for its producer – in this case the architect. The increasing links between computational elements, material elements and project performance and behaviors adds complexity to how we assess project impacts and outcomes. Architects form multidimensional relationships with project elements; we consider the success of an architectural project

in terms of a long-established tradition of valuing form and function simultaneously. How well a piece of architecture performs includes concerns for spatial effect, programmatic support, technical performance and so on. This set of concerns has been expanded in recent times to include broader issues on the project site and beyond. Ecological concerns, dealing with sustainable building practices, material manufacturing and material selection have become increasingly relevant and important. Further, as evidenced by the very theme of this conference, consideration of the ethical dimensions of architectural projects is also broadening how we evaluate and reflect on our work. Issues of access, equality, fairness and inclusion are changing relationships across the lifespan of a project – from preliminary stakeholder representation and consultation to safe construction practices and working conditions⁷.

A similar multidimensional perspective is useful when considering the computationally-oriented project components. Software can be considered in terms of how stable it is in terms of its operation; evaluated in terms of how well the system meets the functional needs of its users; and even aesthetically in its implementation to achieve elegance⁸.

Just as architecture is subjected to a broader set of considerations, software, as an assemblage of algorithms and data streams, is also being studied from social, cultural, political and ethical perspectives. Understanding software from a sociocultural/ethical perspective is on the rise largely in part due the rapidly increasing and diversifying role it plays in shaping our relationship to the world; texts by Carlos Ratti, Adam Greenfield, Marcus Foth, Anthony Townsend, and Ed Finn demonstrate how information communication technologies, through a physical-digital convergence, are having material effects on our built environment⁹. These works disclose the first-order impacts of digital technology as it relates to issues of equity, access, privacy, oversight, security, consumption and other social, political, and cultural consequences. For the most part, such study of digital technology has offered an examination of the *effects* of embedding digital technology, in our homes, cities, automobiles and so on, while not dwelling too much on their *causes*.¹⁰ The technology is seen as a black-box, with almost mystical powers¹¹.

For technology-producing architects, such an obfuscated relationship with our project's computational components is insufficient. As producers of the material assemblages constituting our design work, we need to be literate and possess agency with respect to the social, cultural and political effects across the entire assembly – and this includes not only the material outside the computer, but the material *inside* the machine, and the connection between the two¹². This points to the need for a new set of obligations and methods for understanding and supporting architects' engagement with their projects' computational elements, its *soft materials*, to

support the same level of criticality and engagement as they have with its physical material elements. When the technology is embedded in the material form of a design project and actively mediates performance through time and space the software undoubtedly becomes relevant to external stakeholders and, as a result, these soft materials gain agency in the work – the data/algorithm/code bundles are actively mediating urban life in a similar way to the project's more traditional material manifestations. Thus, the set of lenses through which the projects' digital components needs to be understood expands to include socio-cultural effects.

If we accept both that software is becoming a material used in the production of architecture *and* that we want to build ethically-minded, socio-cultural and socio-political understandings of the impacts of design work, what are the necessary methodologies to support these types of relationships with the *softer* materials of design work? Can the software itself be positioned as a spatial agent in design work? If so, how do we understand this agency? How should we shape architects' relationship with custom software to ensure this agency is acknowledged, managed and leveraged towards the most ethical outcomes?

SOFTWARE AS A SPATIAL SOCIAL-MATERIAL PRACTICE

Reflection on these questions leads to the field of *Software Studies* which focuses on the etiology¹³ of code – studying the role code plays in shaping the effects of digital technologies. Software Studies seeks to crack open the prevalent representations of software as a black box and to understand its inner workings with respect to instructing technologies how to act. Matthew Fuller, an important voice in defining and positioning Software Studies, states that the field “proposes software can be seen as an object of study and as an area of practice for the kinds of thinking and areas of work that have not historically ‘owned’ software or indeed often had much of use to say about it”¹⁴ Work done in this arena has successfully demonstrated that software is a social-material production with profound influence on everyday life.¹⁵

Rob Kitchin and Martin Dodge offer a stream of Software Studies research that we posit is particularly useful in trying to understand how architects can begin to establish relationships between immaterial project components, such as data and algorithms, and real-world project outcomes – be they material, formal, or socio-cultural, socio-political. Their concept of *code/space* captures the idea that “spatialities and governance of everyday life unfold in diverse ways through the mutual constitution of software and geo-spatial practices”.¹⁶ They seek to understand software as a spatial agent by foregrounding the work that software does as a product of people and things in time and space. Code/space occurs when software and the spatiality of everyday life become mutually constituted (ie: produced through one another). If a space is

dependent on software driven technologies to function it is a code/space. An example the authors use to illustrate this definition is a modern supermarket – without a functioning check-out system, a supermarket ceases to have the ability to make transactions and it is therefore transformed into a warehouse until the check-out functionality goes back online. Further, any space that has the latent capacity to be transduced¹⁷ by code constitutes a code/space at the moment of conversion. A useful example here is a café offering an accessible Wi-Fi network that can be transduced into a work space by a patron. This example also suggests an important characteristic code/space in that it unfolds in various ways simultaneously as a result of the interactions and behaviors of the people engaging with it – the café could become a writing studio, an accountant’s office, *and* a remote interview site at any given moment by different people.

Software Studies positions the design and implementation of both software *and* the built environment, and the people that populate it, as constituting mediating and shaping forces of everyday life. It articulates real-world impacts that software has on how space is articulated, defined and transformed: there are environments that simply cease to function if any one of these constituent forces is absent or compromised.

As a corollary, if design and implementation of coded objects¹⁸ is flawed with respect to its socio-cultural considerations, the effects of that error will be transferred out of the machine and into the physical world. A useful, yet cautionary example of this type of transfer is the *Street Bump App*¹⁹ developed by the City of Boston. Conceived as a way in which to help the city locate the more than 20,000 potholes in need of repair each year by passively reporting GPS locations of possible potholes by interpreting an alert smartphone’s accelerometer readings as it moved in an automobile, the app seemed to do what it was intended. Unfortunately, the software had a signal problem: due to socio-cultural factors, namely due to the fact that lower income neighbourhood populations have fewer smartphones and therefore fewer reports, the city’s pothole repair was grossly incomplete, because it was missing input from a significant portion of the city’s population, and effectively diverting services away from populations with access to the fewest resources.²⁰ The app’s code worked well technically, it was easy to use and generally well appropriated by the population that had access to it, yet, it produced significant, unwanted and clearly unethical social effects.

The ethics around code and software are being articulated more and more sharply. One such account is the powerful attribution offered by Stephen Graham that code is used to judge people’s worth, eligibility, levels of access, and ease of mobility – to reinforce social and economic inequalities with little democratic accountability, scrutiny or right of redress²¹. Of course, there are instances where such outcomes are tolerated and even desired²², but as in the Speed Bump example,

it’s easy to see how these effects are inadvertent and even detrimental to planned outcomes and impacts. To minimize the chance of design projects encountering a similar fate, it is essential for designers to integrate questions and criticality around (1) data (sources, format and selection), (2) algorithms (design, implementation, deployment) and (3) coding culture itself²³ in order to confront the spatial and sociocultural agency of their data-driven works²⁴. As we enter new frontiers in computational design, it is no longer enough to only have methods that *integrate* computation and data in design projects but new methods are needed that *engage with it critically* throughout the design process and beyond.

While a robust examination of how each of these components (the data, the algorithms, and the coding culture) contribute to the socio-cultural outcomes of SED work is an ultimate goal, we focus the remainder of this paper on the first element in the triad – data – and look to a discovery and articulation of methods aimed at expanding the degree designers engage with it. The ideas presented here are preliminary and, in alignment with the spirit of this session topic, aimed at positioning and expanding the discourse around computational ethics in relationship to architecture. For us, formulating specific questions around how we understand data, by elaborating on how its quality is defined, examined and evaluated, is seen as a way to help deepen designers’ relationship to its ethical implications.

DATA QUALITY: A MULTI-DIMENSIONAL PROBLEM

Thinking about data in terms of its *quality* offers an entry point to forming a critical assessment of data consumed and produced in SED work. The details of such a process is an active research front rooted in the information sciences but also present in other disciplines such as geography, sociology and computer science. Data Quality (DQ) is often presented as a multi-dimensional concept emphasizing a varying set of characteristics depending on its assessors’ perspective.²⁵ Thus, fitness-for-use is widely adopted as a core principle in DQ²⁶ implying that data quality is relative: its assessment depends on its use and data appropriate for one use may not be appropriate for another.²⁷

There are numerous frameworks which propose a certain set of dimensions by which to consider and examine DQ.²⁸ The framework created by Wang and Strong (WS) entitled “A Conceptual Framework for Data Quality”²⁹ is selected here for closer examination because of its relatively consistent alignment with familiar design considerations such as performance, form, context, and representation. The WS framework is refined to address four key categories and offers a structured guide to organizing dimensions against which designers can form critical relationships with data. (Figure 2)

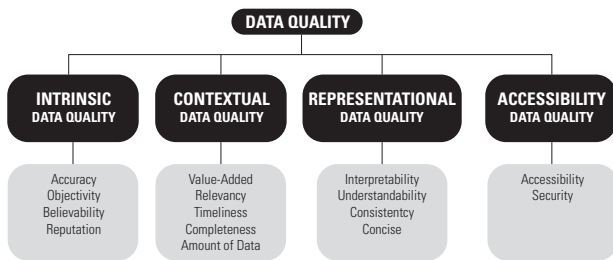


Figure 2: Wang & Strong's Hierarchical Conceptual Framework for Data Quality (1996). The framework is refined to address four key categories: 1.) Intrinsic DQ addresses the extent to which data values are in conformance with actual or true values. This category includes accuracy, the extent data are correct, but also includes objectivity, the extent to which information is unbiased, unprejudiced and impartial. 2.) Context DQ deals with the extent to which data are applicable to the task. Key components of this category include value-added, the extent to which information is beneficial, provides advantages from its use, and relevancy, the extent to which information is applicable and helpful for the task at hand. 3.) Representational DQ addresses the extent to which data are presented in an intelligible and clear manner. 4.) Accessibility DQ concentrates on the extent to which data are available and obtainable.

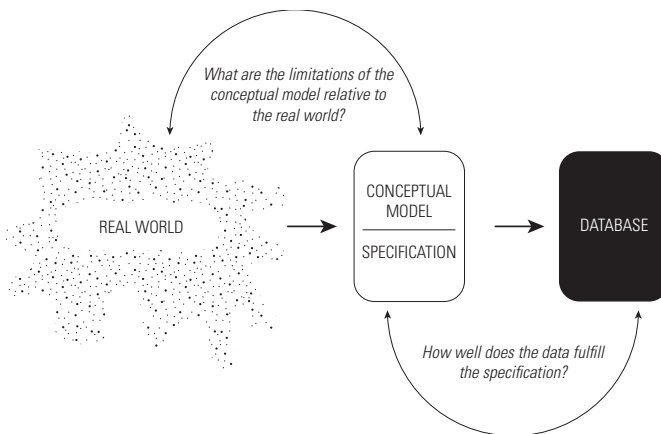


Figure 3: Terrain Nominal. A Conceptual Model as a bridge between the real world and its data-based representation.

TERRAIN NOMINAL: A FOUNDATION FOR ETHICAL AWARENESS

The conceptual organization and terms provided by such DQ frameworks facilitate critical engagement with data integrated in design projects. By exposing the multidimensional quality of data, frameworks identify some of the questions designers need to ask of the data being consumed in their projects.

In order to satisfy a given dimension of a DQ framework, the data must be understood by the extent to which it fulfills the specific dimension. This terminology reminds us that DQ is relative: the extent or degree to which a dimension needs to be met must be specified by the designer. It is only when the designer specifies the degree to which the data needs to be accurate (or relevant, objective, timely, concise...) that DQ

assessment can take place. The sum of these specifications defines the *terrain nominal* which declares, implicitly and explicitly, the required level of abstraction and generalization relative to the real-world phenomena the data attempts to capture.³⁰ Terrain nominal, accepts that there is no objective reality or essential truth the data must capture and instead focuses on establishing a specification corresponding to a perspective of the phenomena which is tuned to the task at hand.³¹ In some cases, data is known to be subjective, incomplete, or inexact; by accepting 'shortfalls' such as these in the conceptual model the data may still be fit-for-use and meaningful to the project.³² Thus, to judge fitness-for-use, the designer needs to not only assess the data against the specification, but also consider the effectiveness and limitations of the conceptual model itself (Figure 3).³³

Terrain nominal is further useful as a conceptual link between the three agents – data, algorithms and coding culture – in shaping the socio-cultural and socio-political outcomes of the embedded software. No matter how simple or complex a coding problem is, or what programming language is used to implement it, the composed code is a manifestation of a system of thought – indirectly articulating a set of values around how real world phenomena are captured, represented, modelled, and processed.³⁴ This is an intense process of abstraction, where the authors of the code decide what gets included and what gets omitted in terms of defining the "world" in which the software operates.

As with any design problem, abstraction is a key process in enabling action; it reduces problem complexity to allow the characteristics, constraints and properties deemed to be most relevant to be addressed, while the ones deemed less important to fade away. It also is the point in a process in which assumptions and biases are seeded for further propagation through software development. Turning to Fuller again, we learn that software can be understood as "a form of subjectivity – the software constructs sensoriums, that each piece of software constructs ways of seeing, knowing and doing in the world that at once contain a model of that part of the world it ostensibly pertains to and that also shape it every time it is used."³⁵

This inherent subjectivity calls on us to be explicit in the assumptions and biases embedded by us in the models used in the software in our projects. These biases may arise for numerous reasons: 1) technical limitations – coders' ability is limited so aggressive decisions around how to represent the problem are made, 2) resource limitations – time and other resources dedicated to examining the problem from stakeholder perspectives and socio-cultural impacts is minimal, and 3) limited self-awareness - default decisions made by coders because of assumptions from their own socio-cultural background or the general culture of the environment around them.



Figure 4: OnTheLine. The project offers a comprehensive framework for enabling the city as a site of participation, thereby offering a mechanism by which to sense the city and enable its strategic development and management. Project & Image Credits: DataLAB, School of Architecture, University of Waterloo.

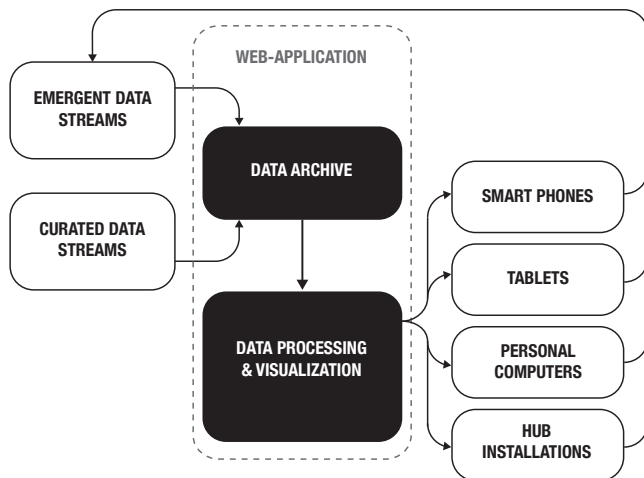


Figure 5: OTL's Datascape. Two primary data streams, curated and emergent, are combined into a suite of physical installations and digital interfaces.

In order to make some of these thoughts more concrete, we use an existing SED project to carry out a post-mortem reflection on how its DQ assessment may transpire.

DQ ASSESSMENT OF ON THE LINE

OnTheLine (OTL)³⁶ was conceived as an exhibition which converts a transit line into a gallery connecting transit riders with the rich collection of destinations located along the route. The project team created a suite of physical installations and digital interfaces aimed at shaping and presenting a collective identity of the region. Utilizing bus shelters, buses, a centralized project display, an interactive website, and a mobile app as media for information transmission, OTL both distributes

and collects information about activities, events, and destinations along the transit corridor (Figure 4). Two complementary sets of data contribute to the unified presentation of destinations: first, the project presents a *curated* and centralized set of local cultural destinations and activities that has been pre-assembled by the project team from a suite of local information resources; second, an *emergent* set of destinations submitted by the public, via social media mechanisms such as twitter hashtags, augments the list of presented destinations (Figure 5). This hybrid data set constitutes a growing directory of local information, engaging a diverse range of participants as project collaborators. The project's various sites, its collection and use of data, and its supporting media and technologies, are combined into a comprehensive framework for enabling the city as a site of participation, thereby offering a mechanism by which to sense the city and enable its strategic development and management.

The following sequence of diagrams (Figures 6-8) presents a high level, intuitive reflection on how we can begin to build richer engagement with the data components of this work. Working together as a set, they seek to: (1) clarify the socio-cultural and technical objectives of the project in order to establish the terrain-nominal, (2) identify and prioritize which dimensions of data are necessary to consider to facilitate a socio-cultural understanding and (3) develop preliminary representation strategies to communicate the assessment of the dimensions in order to build a critical understanding of the project's datascape.

This study of OTL's data represents a starting point for building methods that allow deeper engagement with an SED project's computationally-oriented components. Perhaps most clear in this work is the fact that confronting data quality is a multi-dimensional challenge demanding the articulation of new questions and methods for architects. At this stage, we see value in unpacking these dimensions from an interdisciplinary perspective and mapping them to a design context through

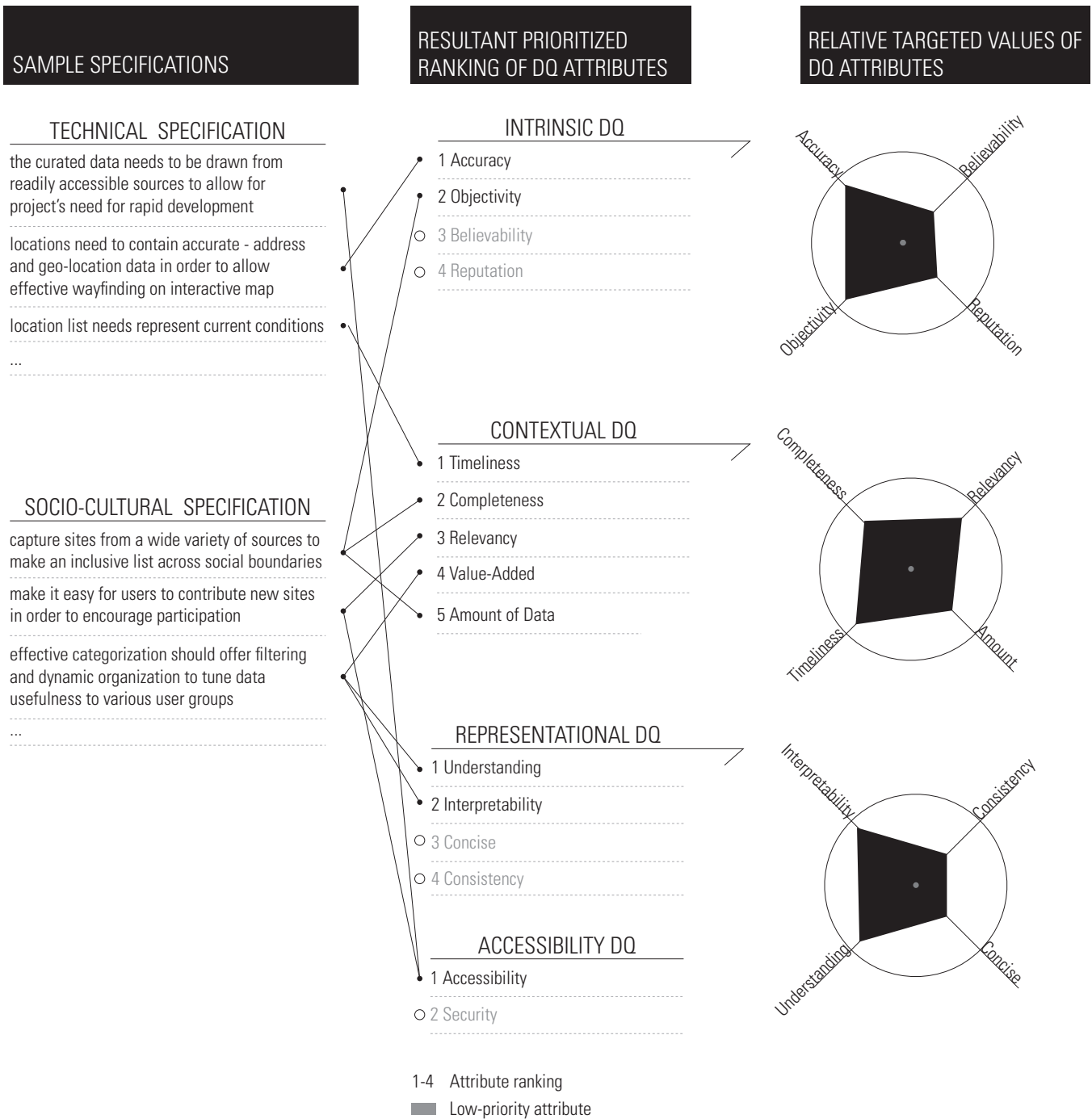


Figure 6: An intuitive definition of the conceptual model and DQ framework for OTL. A sampling of specifications driving the key technical and socio-cultural objectives of the project is offered (left) to establish a conceptual model of the phenomena that the data is meant to capture – namely the name and location of notable sites along the transit line. These specifications establish the key considerations informing the prioritization of the Wang & Strong DQ attributes (center). Connecting lines are drawn to communicate which specification relates to which DQ attribute. A Preliminary intuitive ranking of attributes is also provided. A preliminary set of target value weightings setting the project’s terrain nominal is also presented (right) .

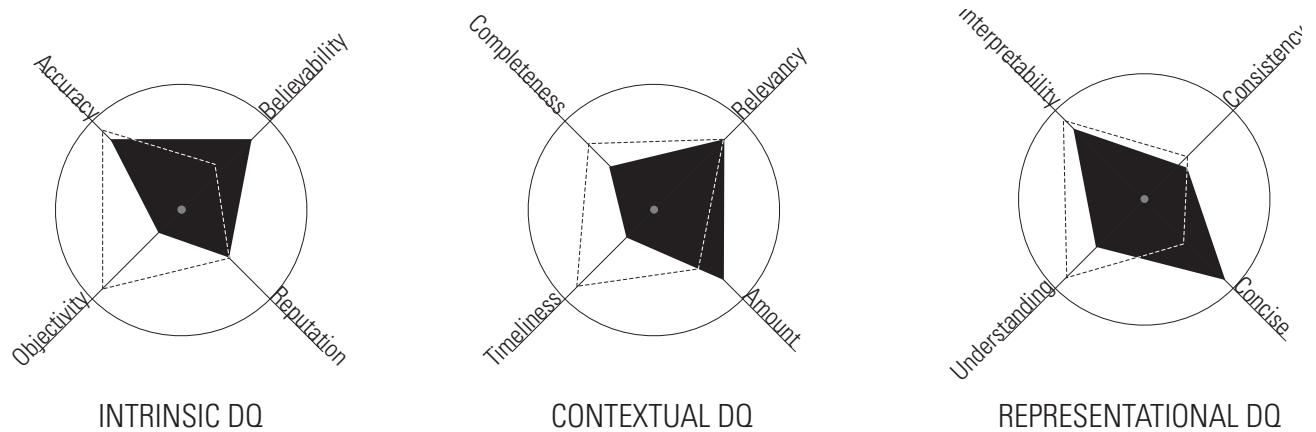


Figure 7: A high-level intuitive assessment of the OTL's Curated Data (black) vs. Target value-weightings (dashed-line). Curated Data was assembled by the project team from a variety of community-level sources, such as the Ontario Heritage Trust and the Waterloo Regional Tourism group, and subjected to manual processing, verification and selection. Processing steps, such as cross-referencing municipal addresses with geocoding results, increased ratings for accuracy and believability. The timeliness attributes received a low rating as some data sources were dated over one year prior to the project implementation and thus increased the probability of venue being closed or otherwise out-of-date. Objectivity was also scored relatively low: the selection process behind the data's inclusion (immediate availability was a key driver and less available sources were excluded) did little to ensure an unbiased and impartial dataset.

the study of a project invested in the social implications of new technologies within the built environment. This technique demonstrates an accessible approach for designers to begin to formulate some degree of criticality around the data they use in their work. The adopted frameworks were effective starting points, but they are biased towards the concerns of external disciplines. The inclusion of these concerns needs to go beyond simply tacking on technical details found in extra-disciplinary literature and instead be tuned in relation to the language, knowledge and methods native to the design discipline. Further reviewing already-existing SED work in order to build an inventory of possible SED-focused attributes is one approach. Surveying SED authors in order to understand existing data-related objectives and values is another. At the forefront of this effort is the realization that all technology comes with a set of assumptions and biases – and, we argue, as a producer you should know about them and control and manage them in a capable way.

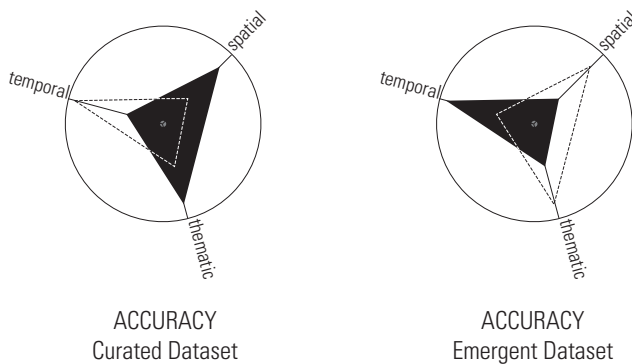


Figure 8: Comparison of relative accuracy of Curated (CDATA) and Emergent (EDATA) datasets. Due to the data's geospatial nature, accuracy is unpacked to include three additional dimensions by which to better understand its quality: temporal accuracy, spatial accuracy and thematic accuracy. Independent examination of each of these sub-properties exposes discrepancies between the two datasets. On the one hand, the real-time features of the EData overcame some of the timeliness issues of the CData discussed above. On the other hand, EData was more susceptible to lower levels of relevance and higher probabilities for mis-categorization resulting in a lower rating for thematic accuracy. Further, users' smartphone location settings caused the precision of spatial information to vary widely across EData entries. Meanwhile, the CData was subject to spatial accuracy verification which ensured highly precise spatial data.

Returning to its larger themes, this paper presents two inter-related discussions aimed at exploring dimensions of technological ethics in architecture. The first one positions architects as producers of technology. The second presents the core constituents of these technologies, understood as data/code bundles, as having a direct role in shaping the socio-cultural outcomes of the technology and the design work. This framing causes us to reconceptualize ethical literacy in SED projects. By including soft materials, such as data and algorithms, in our conception of a project's material assembly they implicitly become a part of the matrix defining the works' ethical dimensions. Thus, an expansion of knowledge and methods to support a more complete engagement with computational components embedded in such work is necessary. We present preliminary attempts at such an expansion by unpacking an understanding of data quality and associated methods for its assessment. This work shapes a potential for architects' relationship with these soft materials so that their socio-cultural agency is acknowledged, managed and leveraged towards ethical outcomes with respect to access, exclusion, freedom, and control. Of course, more

work is needed, not only from the perspective of data, but also from the perspective of its counterpart – the algorithm.

Moving forward, we must remember two things: raw data is an oxymoron³⁷ and software transforms and reconfigures the world in relation to its own systems of thought³⁸. The recent increased use of machine learning as part of our software systems, which can present as a being seamless and neutral, only makes these reminders even more important as these very properties potentially mute the systems' instrumentality as designed socio-cultural agents. Do the socio-cultural values projected by a projects' physical elements match those projected by its computational components? This question and the methods available to answer it are at the core of this work.

ENDNOTES

- 1 Levi Manovich is a media theorist focused on cultural analytics, computational social science, digital humanities, software studies, media theory, and visual culture. He is the director of the Cultural Analytics Lab at City University of New York (CUNY). This quote is attributed to him in: Kitchin, R, and Dodge M. "Code/Space. Software and Everyday Life". Cambridge (2011) Print. 246.
- 2 Peters, Brady. (2017,08,15). Project Distortion II. Retrieved from Brady Peters: <http://www.bradypeters.com/project-distortion-ii.html>
- 3 De Monchaux, Nicholas. "Local Code - 3,659 Proposals about Data, Design & the Nature of Cities". New York: Princeton Architectural Press, 2016. Print.
- 4 Beesley, P. (2017, 01, 10). "Sentient Chamber". Retrieved from Philip Beesley Architect Inc.: <http://philipbeesleyarchitect.com/sculptures/Sentient-Chamber/Sentient-Chamber-PDF-Article.pdf>
- 5 Beaumont, J. R. (2016, September 27). "The Palimpsest". Retrieved from Palimpsest: <http://www.johnrussellbeaumont.com/the-palimpsest>
- 6 Research groups focusing on these questions include: The Living Architecture Systems Group is an international research partnership probing the potentials of architecture's capacity to integrate living functions (<http://livingarchitecturesystems.com/>). The Centre for Architecture and Situated Technologies operates out of the Department of Architecture and Planning, University at Buffalo and is focused on the evolving and growing implications of new technologies within the built environment from social, political, ecological, and material perspectives. (<http://cast.b-ap.net/>). The Interactive Architecture Lab is based in the Bartlett School of Architecture at University College London interested in the Behaviour and Interaction of Things, Environments and their Inhabitants. (<http://www.interactivearchitecture.org>)
- 7 Horton, Guy. "How Migrant Workers are Silent Collaborators in Global Architecture: Beyond Qatar, Fantastic Architecture is Dependent on Migrant Labor;" *Metropolis*. March 2014. Web. August 2017.
- 8 Fuller, Matthew. "Elegance." in *Software Studies: A Lexicon*. Ed. Matthew Fuller. Cambridge Massachusetts: The MIT Press, 2008. 87-91. Print.
- 9 Finn, Ed. *What Algorithms Want: Imagination in the Age of Computing*. MIT Press, 2017. Print. Foth, Marcus. *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City: The Practice and Promise of the Real-Time City*. IGI Global, 2008. Print. Greenfield, Adam. *Radical Technologies: The Design of Everyday Life*. New York: Verso, 2017. Print. Ratti, Carlo, and Matthew Claudel. *The City of Tomorrow: Sensors, Networks, Hackers, and the Future of Urban Life*. Yale University Press, 2016. Print. Townsend, Anthony M. *Smart Cities: Big Data, Civic Hackers, and the Quest for a New Utopia*. WW Norton & Company, 2013. Print.
- 10 Kitchin, Rob, and Dodge, Martin. *Code/Space: Software and Everyday Life*. (MIT Press, 2011), 245.
- 11 Finn, Ed. *What Algorithms Want: Imagination in the Age of Computing* (MIT Press, 2017). Print.
- 12 De Monchaux (2016).
- 13 Etiology is the investigation or attribution of the cause or reason of something.
- 14 *Software Studies: A Lexicon*. Ed. Matthew Fuller. Cambridge Massachusetts: The MIT Press, 2008: 2
- 15 Kitchin and Dodge. (2011): 3-21
- 16 Ibid, 16
- 17 Kitchin and Dodge understand space as constantly being in a state of non-deterministic becoming, operationalized through the process of transduction. They offer an expanded genealogy of thinking about space to position their specific appropriated view. (2011, 65-80)
- 18 For Kitchin and Dodge, software is embedded in everyday life at four levels and produces coded objects, coded infrastructures, coded processes and coded assemblages. Coded objects are reliant on software to perform as designed (credit cards, electronic alarm clocks, smart thermostats). Coded infrastructures can link coded objects together but also include infrastructures that are themselves, to any degree, monitored and regulated by software (transportation systems, HVAC systems, and the network of components in an individual contemporary automobile). Coded processes deal with transactions as flows of digital data across coded infrastructures. Finally, coded assemblages occur when multiple coded infrastructures and their constituent coded objects converge to produce a particular environment.
- 19 <http://www.streetbump.org/>
- 20 Crawford, Kate. "The Hidden Biases in Big Data." *Harvard Business Review*. April 2013. Web. <<https://hbr.org/2013/04/the-hidden-biases-in-big-data>>.
- 21 Graham, Stephen DN. "Software-Sorted Geographies." *Progress in Human Geography* 29.5 (2005): 562-80. Print. Cited in Kitchin and Dodge (2011): 105
- 22 Adam Greenfield (2017) unpacks numerous ubiquitous technologies, such as smart phones, and presents their wide-ranging effects in reshaping our economy and politics while simultaneously having profound impact on ethics, security and privacy.
- 23 There are many conversations currently occurring around the disproportionately male-dominated culture behind many tech start-ups and projects. One such account was offered here: McMahon, Tamsin. "What's Behind the Tech Industry's Toxic Masculinity Problem? Inside the Valley of the Bros." *The Globe and Mail*. July 22, 2017. Web. Accessed July 25, 2017, <https://www.theglobeandmail.com/technology/toxic-masculinity-in-silicon-valley/article3759481/>.
- 24 From the perspective of Software Studies, these three components are the primary drivers behind the black box.
- 25 Klein, Barbara D. "User Perceptions of Data Quality: Internet and Traditional Text Sources." *Journal of Computer Information Systems* 41.4 (2001): 9-15. Print.
- 26 Wang, Richard Y., and Diane M. Strong. "Beyond Accuracy: What Data Quality Means to Data Consumers." *Journal of Management Information Systems* 12.4 (1996): 5-33. Print.
- 27 Tayi, Giri Kumar, and Donald P. Ballou. "Examining Data Quality." *Communications of the ACM* 41.2 (1998): 54-7. Print.
- 28 Knight, Shirlee-Ann, and Janice M. Burn. "Developing a Framework for Assessing Information Quality on the World Wide Web." *Informing Science: International Journal of an Emerging Transdiscipline* 8.5 (2005): 159-72. Print.
- 29 Wang and Strong (1996).
- 30 Veregin, Howard. "Data Quality Parameters." *Geographical Information Systems* 1 (1999): 177-89. Print.
- 31 Chrisman, Nicholas. "Development in the Treatment of Spatial Data Quality." *Fundamentals of Spatial Data Quality*. Eds. Rodolphe Devillers and Robert Jeansoulin. (London: John Wiley & Sons, 2010), 21-30. Print.
- 32 Veregin (1999).
- 33 Veregin (1999).
- 34 Kitchin and Dodge. (2011): 28.
- 35 Fuller, Matthew. *Behind the Blip: Essays on the Culture of Software*. (New York: Autonomedia, 2003). Print: 19. Cited in Kitchin and Dodge(2011): 27.
- 36 OnTheLine was developed by DataLAB at the School of Architecture at the University of Waterloo in 2014. Project Leads: Mona El Khafif and Maya Przybylski. Student Design Team: Zak Fish, Lea Koch, Daniel Malka, Thomas Nouissis and Jake Read.
- 37 Gitelman, Lisa. 2013. *Raw Data is an Oxymoron* (MIT Press).
- 38 Kitchin and Dodge. (2011): 26.