Progressive Technology Network

December 8, 2016

1 Overview

At their best, science and technology serve the people: innovation ensures that nutritional food, safe living quarters, effective health care, and access to education and information are available to all. When the basic affordances of a productive life are not scarce, they are decommodified; they cease to become markers of privilege or status, things to compete over. Science fails whenever a person has to confront life without adequate food, shelter, knowledge, and well-being.

At their best, science and technology also serve Democracy. The deliberative, evidence-based ethos of science and mathematics, technology and engineering, reveals a miniature version of the governmental rationality that humanity must demand across all states and sovereignties. Political platforms are to be defended with reason and argument, not coercion or physical force. Attempts to impede the free flow of ideas — by imprisoning journalists and academics, for example, or even denying them forums for writing and communicating — such actions are violations of human rights, but beyond that. They are desecrations of a Democratic contract that unifies people across race, ethnicity, language, religion, and nationality.

We know that science has not always lived up to its responsibilities. And science is not a central topic for the Academic left, and at best dutifully but casually solicited by the progressive political community. Progressives understand the importance of science on Climate Change, they perceive in some forms of technology an opportunity for a more sustainable civilization, and they condemn the irrational or manipulative obscuring of science for profit or power. But progressive platforms appear, more often than not, to lack scientific detail and precision.

This is a major oversight. Science and technology is intrinsic to a progressive view of the future: an unstated assumption of policies favored by the global Left is that science can unlock resources and channels for communicating goods and ideas which will alleviate the ugly imbalance in our species between rich and poor, between First World and Third. Science is implicit in the hopes for resettling refugees, for pro-Democracy activism, for rising standards of living worldwide. It is science, for example, which will ensure that Western nations have enough plentitude of food, housing, and medicine to absorb cities ' worth of refugees from Syria and other conflict zones. It is technology that will provide educational resources so that those refugees do not live on the margins of society. It is science and mathematics which provide our firewall against environmental degradation caused by the just demands of people around the world for elements of Middle Class life — air conditioning, clean running water, cars or public transportation, modern health care. Progressive leaders must take these challenges seriously. An intrinsic part of Progressive politics must be identifying technical policies with considerable detail, in contributions from computational, natural, and social sciences, how the progressive vision of a more equitable society can be realized in practice.

Progressives cannot assume, in short, that deep social justice will seamlessly follow from a change in reigning ideology; that the only factors hindering an end to extreme poverty or inequality, or racism and ethno-national isolationism, is the prejudice of voters or the charisma of demagogues. True progressivism — notwithstanding critiques of particular policies, like the Trans-Pacific Partnership — is globalist at heart, driven to promote the justice and equality, hopefully advanced by enlightened policies, to communities irrespective of race and place. Multiculturalism is a logical outcome of international openness, the erosion of borders and walls, but it is also a reasonable path to collective well-being, as neighborhood solidarity builds on a foundation of education and infrastruture and small-scale entrepreneurship rather than homogeneity, and people are free to settle in locales ideal for their talents and aspirations.

But to envision a benign internationalism which encourages rather than constricts the flow of people and ideas — the futher transcendence of the nation-state and pluralization of communities large-scale and small — this vision can easily be seen as at best idealistic, at worst naive. Progressives cannot think globally without also thinking locally. Progresssivism must be made to work, materially and concretely, because the infrastructure of modern life can be made to work, for a broader range of the human community than enjoys it now. Brick by brick (and not only metaphorically), modernity cannot be left vulnerable to the randomness of political power. Progressives must demonstrate, even if on small scales, the policies they want governments to adopt on large scales. Where they are not in government, or do not hold legislative majorities, they can be a shadow government, an unofficial civic organization. Progressives need to show a nervous public that diversity — wedded to a scientifically backed educational and entrepreneurial construction of the local neighborhood, its economic and physical operating — yields the security and stability that ethnocentric retrenchment falsely promises. For decades Progressives have delivered ideas; now Progressives must deliver pipes, wires, bricks, and vegetables. But we must do this in a way that shows how science and community solidarity, not market ideology or ethnocentrism, can unleash the potential of modern societies to provide essentials like food and health care more effectively.

The presumption of scarcity is as irrational as the presumption of abundance. We have seen a worldview that presents itself as prudently and pragmatically accommodating scarcity — and propagandizing a system of values and mores that are supposed to empower communities in the face of limited resources, values like work ethic and long-term planning — we have seen this ethic mutate into an irrational acceptance of poverty and injustice. It is irrational to assume that hunger is inevitable, for example, without calculating how much food is actually needed and how much is produced. Acquiescing to people being hungry even when such calculations show that hunger is unnecessary, represents a cynical decay of values like prudence and industriousness: using hunger and poverty punitively, essentially allowing societies to hoard resources so that citizens cannot take provision for granted. We can see this in the backlash against provisions for immigrants and refugees (modest though those are), as if communities feel that offering such supports denies them a capacity to use poverty and degradation as a threat against would-be community members, a message that acceptance into communities is guarded and privisional. People may feel that outsiders joining their community is a privilege, and should be contingent on their accepting communal norms. Communal norms should not be casually ignored, but such coercive homogenizing is a slippery slope that can easily go terribly wrong.

We should be sensitive to the danger of this coercion whenever societies seem to casually assume and accept limits on infrastructure — that certain programs are impractical, certain people should not be granted inclusion, and so forth — whether these concerns are voiced in overtly racial language or hidden behind euphemisms of “small government”, “tradition”, “populism”, or even “pragmatism”. It is Science which should be the arbiter of what is practical and what is not, hopefully without bias toward

race, creed, or any other group marker. Modern science and technology give communities resources to expand, in terms of size and also in terms of their diversity and culture. As diasporic communities spread around the world, more and more people take on layered identities, connected by threads of language and culture to others in remote places, but also to their neighbors and neighborhoods, and to their peers in interests and workplace. With these multiple layers, each person's understanding and experience truly is unique. This is a positive virtue of diverse, welcoming communities. It is also one that science and technology should commit to reinforcing — partly by ensuring that science provides a common discourse and problem-solving recourse that transcends cultural difference, and partly by exposing the lie that exclusion and suspicion is needed for communities to prosper.

This is not really suggesting a change of course: the Progressive vision, "deconstructed", reveals an implicit faith in science and technology which is largely unstatated and, as a result, has not been acknowledged in the bedrock of Progressive platforms even as it stands as a bedrock of policy. Computational tools are an important part of this bedrock. Advances in food science and agriculture, in manufacturing and industrial production, in energy and sustainability, make it more efficient to provide nutrition, energy, and goods to a larger public — which allows modern societies to expand their population without diminishing their standard of living. But all of these developments are haphazard, weaving through a landscape of competing interests and conflict between government- and market-oriented ideologies. The modern world's plentiful supply of goods and consumer products has not eliminated poverty, even in modern societies; like a key which only certain people can turn, modern production is held back and afforded to people only provisionally, through cultural assimilation and the workplace. People can disagree on whether or not society's criteria for who may join the community, as citizens and middle class workers, are fair; but the very fact that the question exists shows that modern production promotes collective flourishing only relative to and within the boundaries of society and politics. So while science serves society by making production yet more efficient, this by itself does not follow a path directly from innovation to egalitarianism, from efficiency to inclusiveness.

There is, however, an added dimension to scientific progress; a computational infrastructure which makes individuals more productive, not just factories and farms. Computer tools also make research quicker and easier — in the specific realm of science and politics, consider the potential of searching political and government-info APIs specifically for science-related content (legislation, lobbying, campaign spending by companies whose actions have scientific implications, uncovering scientific funding itself, etc.). Computer tools also make civil logistics quicker and easier. In the information age, at least potentially, the civil servant, the charitable volunteer, the nonprofit employee, the community organizer, has a more powerful engine at their disposal: the information they need and the communicating they rely on can be created or accessed more quickly, their tasks and organizing taking less of their time, their collective knowledge more widely disseminated and combined.

This computational dimension of modern science and technology makes a scientifically driven vision of modern progress fundamentally different from its industrial-age predecessors which, in the eyes of many, have been discredited. Computational tools are among the most flexible and accessible. Whereas nutritional policy is tied to farms and agriculture, health policy is tied to hospitals and pharmacology, education is tied to schools and colleges, and so on, the computational tools for progressive government truly rise up from the grass roots. Open-Source software can be developed and reused in many governmental and activist contexts. Data Sharing can help organizations interconnect and share resources. Quality software can speed up organizing and logistical chores, making activism and civil society more efficient. This does not demand institutional coordination or bureaucracy: any single person can develop significant software components which others will piece together into real-world applications.

This web site organizes a few platforms that aspire to join a civic-computational ecosystem. First and foremost, I will gather here code libraries and examples which I believe can be part of the solution, for a network of technologically sophisticated, forward-looking progressive individuals and organizations. These libraries work with cutting-edge tools like Docker (a cloud hosting system), Clasp (a Lisp dialect), and Qt (an ever-expanding cross-platform application development framework). This site (via an associated Github page) currently hosts tens of thousands of lines of computer code providing capabilities related to Application Programming, Document Generation, Qt Scripting (specifically a Clasp/Qt interface), Data Management (with an emphasis on XML and XQuery data and Object serialization), and Web Hosting (including a new C++ HTTP server engine). To browse and preview some of this code, please take a look at the "ScignScape" repositories here. A lot of this code is, in all truthfulness, relatively unpolished; for the most part it is code directly used to create documents and pages for this site and its back-end. I will gladly work with scientific, cultural, and/or activist communities who may be interested in adapting some of these code libraries to their own needs. I will also certainly include links to other useful technologies, as well as to other groups, web resources, technical articles, or any other material which others may deem somehow relevant to science and technology in a progressive political context — suggestions are welcome.

This site also includes links to hundreds of research papers and articles related generally to the intersection of science and society, science and politics, science and humanities, and computational matters. The fields covered by the articles gathered here generally relate to the similarities and differences between scientific and humanities viewpoints; the position of contemporary computer science as straddling this relationship; formal computational models of humanistic phenomena like language and visual experience, or computational tools for human concerns like health and education; humanistic or philosophical perspectives on science and technology, such as Philosophy of Science and critical media studies for the internet age; and the intersection of science and politics. Again, it would be great to expand the collection of materials tracked here with other papers, or other resources like news and event APIs. I would also like to publish articles with extra annotations and formatting for software analysis and curation — better tools for interactive display of articles and embedded data which cah further research and textual analysis. For an example of the kind of interdisciplinary research I believe is important, readers may want to browse my article *Cognitive State Semantics and the Condition of the Sign*, which discusses Link Grammar, Dependency Grammar in general, Type-Theoretic Semantics, Cognitive Linguistics, and the Philosophy of Science. A couple of my other narrower papers on Computer Science are also available on the Documents page.

The original articles and the web pages here are generated with a custom markup langauge that compiles to HTML, LaTeX, and Lisp code. The "Lisp" representation is actually an extended version of S-Expressions that captures certain relational data structures in document text; in principle it can be used to notate semantic graph format data in domains like Semantic Web, Dependency Grammar, and Conceptual Graph Semantics. Collectively, I refer to the document preparation tools made available here as the "ScignScape" library, which includes C++ and Lisp code. One feature is cross-generating HTML and PDF files; for a PDF file generated from the same sources as the web pages on this site, click here. Translating from HTML (via LaTeX) to PDF is not entirely straightforward, and the document will not look good without LaTeX commands being properly defined (the above-linked version of the pages does not do much formatting, since it is mostly for illustration). The documents and pages are written in a format called NGML, and can be converted via C++ to HTML or PDF documents, or else to the "Khi-Forms" S-Expression variant where a Lisp library can then do a final conversion to (say) HTML (thereby allowing Lisp code to produce values inserted into the final text). I provide NGML and KhiForms as potential languages for document preparation and serialization, respectively. Links to view the NGML and KhiForms version of each web page will generally be

available at the bottom of the page.

The components discussed here are under development, and will need modification to work for a range of real-world use-cases. But I hope they serve as an example of what is possible with a technology that is built from the ground up around principles of rigorous data modeling with a goal to data sharing, Semantic Web principles for document curation, and a commitment on the part of civic-minded organizations to provide a modern information ecosystem, for example developing data feeds and APIs as well as websites. Docker provides a powerful foundation for free web hosting with C++ server code that allows complete control of HTTP networking (one small example is automatically inserting the non-HTML representations of web pages as links into the pages themselves). There are many small details where a custom web environment can help organizations tailor data and documents to their needs and their vision for interacting with their communities and their peers. Hopefully the tools published here can show some ways of unifying these small details into an overall functionality that continues to empower science and technology to serve its proper societal place.

2 Web Hosting

Web hosting and development is one of the basic needs of any organization, including activist communities. Fortunately, relatively new Cloud Hosting options allow web sites, which may certainly be adequate for many situations, to be hosted free of charge. This is preferable even to low-cost hosting options — even if web hosting constitutes only a fraction of operating costs, the logistics of making or keeping track of automated payments can be a nuissance, especially in a volunteer-driven community. It is an unfair burden if responsibility for keeping payment streams up-to-date falls on one person. Also, organizers may be reluctant to share their financial information online, which can count against some hosting solutions even if they provide an unpaid "basic tier" service.

Organizations may also want to steer away from free or low-cost hosting which exercises some control over content, such as including embedded or popup ads, or limiting the kind of programming a site can support behind the scenes. A new group may want to get a good site up quickly, and may not perceive a need for sophisticated back-end coding or data management. Nevertheless, their need for more sophisticated software may increase as the community expands. The more that a group controls its own technology — deferring as little as possible to other companies, including for example social networking — the more it can craft content and practices in accord with its values. Open-Source social media platforms, for example, like Pump.io and Diaspora, provide an alternative to large commercial interests, that groups can install and run on their own server space. Using bare-bones cloud solutions filled in with custom software is also an alternative to platforms like Google Cloud or Amazon Web Services, which provide more capabilities "out of the box" but leave sites dependent on corporate interests, some of whose policies they may disagree with.

Contemporary Cloud technology gives communities the option of developing web sites and other web resources with much greater control over content and development — with the caveat that the requisite computing environments can be difficult to master. These solutions are not, in general, first and foremost Web Hosting providers; instead, programmers adapt multi-purpose Clous environments — often called "Platform as a Service" (PaaS) — to serve web content. A PaaS environment is a kind of cloud-hosted computing space which mimics a single (typically UNIX-like) computer. To publish a web site, the proper web server technology can be deployed (or built from scratch) running as one program in that environment, supported by the file system and perhaps secondary processes, like database engines. In such a setup the production server environment will generally look very similar to

what is seen on testing and development computers, minimizing deployment and site management.

One variation on the PaaS idea, that has recently garnered wide praise and attention, is *Docker* — a technology that "packages" environments into (mostly) self-contained "images" that can be reused and shared. A web site compiled as a Docker image can be quickly uploaded to a Cloud Server and can migrate between hosting environments in minutes. Docker code is designed for reuse, with images being built on the "basis" of prior images; as a result, organizations can share Docker code collectively and help each other get a head start to deployment. Docker images are hosted on Cloud servers via *containers*, and a variation on PaaS which supports Docker hosting is often called "Container as a Service" (CaaS). So one option for building a modern, freely hosted web site is to register with a free CaaS provider. There are several options here, although only a few of the existing CaaS companies seem to provide truly free, currently available, no-strings-attached Container hosting. One which deserves mention (vith caveats to be discussed below) is OpenShift, from rhcloud, which may be the best and most sophisticated option going forward. Another good option, especially for getting something started quickly in the near term, is a Japanese company called Arukas. Cloud suppliers like OpenShift (which provides PaaS and Caas options), Heroku (which provides a popular PaaS environment), Arukas, and other Docker/CaaS hosts, enable a new generation of sophisticated non-commercial web development. These free services provide a more comprehensive development environment than other options for getting a website up and running at no cost. On the other hand, taking full advantage of such next generation technologies requires study, effort, and/or expertise. The current ScignScape distribution is at least a start in that direction, and perhaps the ScignScape libraries can spur discussion about strategies for leveraging the ecosystem combining tools like Docker, Qt, and OpenShift. This is why a self-contained, open-source, quick-install web hosting framework can be a valuable resource. The current ScignScape distribution is at least a start in that direction, and perhaps can spur discussion about strategies for leveraging the ecosystem combining tools like Docker, Qt, and OpenShift.

The mostly self-contained nature of native-compiled Docker images is a significant benefit. Since Docker images are themselved cloud-hosted on sites like Docker Hub, the images can be readily redeployed on multiple machines. Moreover, with a bit of attention to how the execution environment is set up, the source code and auxilliary files, from which the Docker imags are created, can also be packaged and compied from person to person. This is ideal in a situation where multiple people want to work on a web site in a decentralized way; and where communities want to help each other out, by sharing tech strategies and solutions. ScignScape promotes these possibilities by working within a Qt environment, including Qt Creator as an Integrated Development Environment (IDE). Each web site, in this strategy, corresponds to a distinct Qt project which can be built and run inside the IDE, so creating a development version of the site (running on "localhost") is almost automatic. Customizations for each site can then be incorporated with new C++ code, and while C++ is not very often used for relatively simple sites — in the Web context, C++ is mostly employed when extensive numerical computation is called for; Google for example has published a noteworthy collection of C++ libraries derived from their internal analytic engined — C++ does facilitate excellent IDE support, which offsets any extra complications in the language itself. Debugging, type-specific Front End design, and other development tasks are often easier with C++ IDEs than with more popular web development languages, like Ruby or Python. For example, with a native HTTP server bundled in to the web platform — and also database support — there are no issues with dependencies, "gems", and so forth; the programming supporting a site truly is self-contained. The Docker image code can be copied from one person to another with the recipient able to run and edit the site directly in Qt Creator.

Another advantage to self-contained Server projects is the option of alternative protocols in place or on top of HTTP. There are various reasons why developers may wish to extend or replace HTTP

— one reason is that HTTP is a very comprehensive protocol of which only a select group of features is widely used in web development. Indeed, accessing web sites through conventional browsers places restrictions on the kinds of HTTP requests that can be generated, This limitation can actually be used productively when implementing non-browser Front Ends — with fine-grained control over the exact content of HTTP requests, front-end code can send requests which are structurally different than the kind of requests generated by users browsing the site, and this can be used on the Server to support these front-ends differently. So developers are free to exploit lesser-known features of HTTP to create specialized protocols specific to custom front-ends and APIs, to be used "in-house" or among a collaborative community of users who are able to coordinate specialized data-sharing solutions. This can include advanced security protocols, like Quaternion or Elliptic-Curve Cryptography, although the properties of these protocols should be clearly understood when sites decide to host or receive sensitive data where Cybersecurity becomes relevant.

Meanwhile, with respect to web protocols, another possibility goes in the opposite direction — instead of adding to HTTP by taking advantage of its more complex features, there are numerous cases where alternative protocols are desired which are simpler and more light-weight. Since many obscure aspects of HTTP are rarely used, there are many use-cases where HTTP itself can be replaced with a simpler protocol that is more narrowly defined around the content being transmitted. Feasible alternatives include UDP (User Datagram Protocol) and CoAP (Constrained Application Protocol), which is emerging as a popular protocol for the "Internet of Things" (IoT). For security reasons, experts have raised flags about objects and devices using the same protocols as regular computers. Since most objects work with very limited kinds of data (such as measuring or setting basic physical properties, like temperature), it is more secure to restrict IoT Networking capabilities to requests and operations specific to the objects' narrow data models.

Even outside the Internet of Things, some programmers have also advocated for more constrained and targeted Networking paradigms in general, both for security and efficiency reasons. This goal overlaps with "Green" technology: Networking and Data Warehouses is a very energy-consuming sector, contributing (by some estimates) up to two percent of global "Greenhouse Gas" emissions (roughly equivalent to the airline industrry). Reducing this footprint by shrinking the length and quantity of Web Requests can have a nontrivial impact on the global environment. As a result, some advocates have envisioned a fundamentally different architecture for the Internet which prioritizes transmission efficiency, whereas Web Frameworks since the emergence of "Web 2.0" have clearly gone in the opposite direction. Important, related initiatives include the "Karlskrona Manifesto for Sustainability Design" (see sustainabilitydesign.org). Some sites are "green certified", meaning that site owners pay "carbon offsets" based on approximations of the energy used to host and serve the site's content. Aside from the environmental benefit (symbolic as well as actual), simply preparing for "green" certification helps developers focus on computational efficiency, which has many side benefits, like making sites easier to manage and join in with data sharing networks.

Moreover, even after a web page has been loaded, modern design often calls for frequent back-and-forth data transmissions between browser and server, as sites rely extensively on JavaScript and transfer formats like AJAX (Asynchronous JavaScript and XML). These techniques can make sites feel more modern and response for visitors, but web design which does not consider transfer efficiency can have many negative consequences, including slow and unresponsive User Experience (UX) overall, as well as contributing to an ingrained Internet wastefulness. This is a reason for using basically inefficient techniques like AJAX and JavaScript-heavy web design judiciously, if at all, and for prioritizing "Sustainability" on both Server and Front-End code. The total energy consumed by a single web page is a product both of the number of bytes transmitted when the site is first displayed,

and then by the "secondary" data transmitted by protocols like AJAX. Potentially, both of these quantities can be significantly reduced using Front Ends optimized for particular sites. This may be an additional motivation for members of a group to collectively use a customized networking application in lieu of (or in addition to) a conventional web application.

Most people who go online underestimate the quantity of data transmission involved in even basic web browsing. Web pages usually piece together content from a plethora of images, scripts, styling CSS files, and often ads and plugin content like videos and "sponsored links". Seen just in terms of sustainability, the energy used to send tangential content in many cases dwarfs what is needed to provide the inforation which visitors actually want to see (this is an often unacknowledged problem with clickbait-driven business models, one of many). Unfortunately, sometimes even progressive-minded organization seem to fall back on dubious "secondary content" practices, perhaps to recoup the expense of building web sites to begin with — which is understandable but perhaps can be mitigated with better development tools and inter-community collaboration. Since most sites' goal first and foremost is to attract visitors, understandably developers focus particularly on pages' visual appearance. But while web pages are indeed a visual forum — the web is a visual media, at least in part — pages are not *just* visual content. Web pages are less strictly visual than PDF file, for example, which (even though they can also include annotations, forms, hyperlinks, and other "structured data") are internally modeled as graphical images. PDF files maintain their layout and pagination however they are visually resized ir viewed from different devices. Web pages, by contrast, can have significantly different layout and appearance on different platforms. Any web page is defined by the information it presents as much as by its visual appearance, and it is better not to design a site primarily around looking good on one single platform. Without completely disregarding visual design, emphasizing pages' important content over and against their visual appearance helps sites serve their visitor's needs, and it can also be better for the environment.

The above points lay out some advantages of a self-contained, finely controlled Server environment enabled by technologies like CaaS, where an organization (or federation of organizations) can potentially use a fully customerized Server application, with their own Server, Front End, and Networking protocols and implementations. This development can proceed in stages, starting with the installation of a basic Server executable. The SignScape repository on GitHub, provisional though it may be, is an example of a code base that can be used to initiate this development. After ensuring that the Server runs properly in several environments, programmers can then consider the nature of Front-Ends they wish to support — ordinary web browsing, APIs, and/or custom-built Client Applications for site maintainers, internal use, and/or the general public (the latter option requires a more careful consideration of security and cross-platform support).

The SignScape "Server" framework provides HTTP capabilities that can be run in several ways. For development, the server can be run as a standalone C++ executable; as just mentioned, the code can be run from inside the Qt Creator IDE or (with proper load library paths) run directly as an executable. Alternatively, the server can run as a local Docker service: the Github source includes several sample bash scripts showing how the executable can serve as a Docker image. In this guise, SignScape Server adopts the popular design of packaging native code via Docker by basing the image on a Linux platform (in this case, Ubuntu). In other words, SignScape Server gives you a cloud-ready general Ubuntu environment, within which the server program is just one executable. The advantage of this setup is full control over the libraries and assets which the server can use — for example, the most up-to-date Qt versions. In short, this tactic brings a full Qt environment to Server-side Development; this is especially useful for designing integrated solutions with custom client front-ends and server back-ends designing in consort, with similar types and code base on both sides.

With respect to OpenShift in particular, this is also an advantage over the older OpenShift platforms (though as of November 2016 it should be noted that the OpenShift "NextGen" platform, which uses Docker, is experimental and only available for limited previews).

Uploading to a CaaS platform represents the third option for running SignScape Server. Once again the Github distribution offers sample bash scripts to set up a system for modifying and updating the Docker images (called "Pods") within OpenShift. OpenShift NextGen is a complex platform and working through the intersecting concerns of Pods, Deployments, Shared Volumes, Services, and Routes takes getting used to. The basic steps are: push the Docker image to a repository; create an OpenShift application; register an OpenShift service; create one or more OpenShift routes (this step can be done via the NextGen web portal); create a Persistent Volume Claim; and, when updating the application, it may be necessary to scale the Pod down to zero deploymets before using update-image, then scale up again. Most of these steps are modeled by bash scripts in the SignScape "Docker" repository, which give a sense of code that can make them work, though need to be adjusted for different projects.

Working with "Persistent Volumes" is one of the more difficult aspects of CaaS programming. The data in a Persistent Volumes is retained in between "Container Restarts", whereas other data is erased when a Container-hosted Web Site is updated and recompiled. Numerous companies appear to provide Persistent Volumes only on paid subscriptions, even if a basic CaaS platform is freely provided. In this situation, developers have a few options, including staring persistent data in a format suitable to be copied to a temporary location during a restart, and then reconstituted; or designing the server to allow many changes to be made without an actual restart. The simplest Docker server can be just recompiled and restarted whenever a change is made, even a minor change like editing one single web page. This is perfectly good for simple (or even moderately complex) sites, so long as persistent data management is resolved. For many sites, all data may originate from sites' creators and maintainers, and making that information available on web servers is simply a matter of sharing it with the general public. Complications arise when some data is instead dynamically created by site visitors, for example through web forms, or when a site is designed as a decentralized platform to which many individuals may contribute. In that case a server-side database will store material that does not exist elsewhere, without maintainers deliberately copying it, and this material is subject to erasure unless it is stored in Persistent Volumes. For these scenarios server programs should be designed so that they rarely need to be restarted and can readily back up their data — especially if Persistent Volumes are not freely provided.

The long-term goals for SignScape include making this aspect of development easier, by providing remote scripting capabilities via Lisp (or, hopefully, Lisp code generated by the experimental R/Z language). This would allow remote edits like adding and modifying web pages, embedding content and instructions in encrypted HTTP messages. Access to SignScape sites can be provided by custom-designed management software, which would generate the proper scripts corresponding to maintainers' needed functionality, like editing web pages and creating local backups of server-side data.

The techniques outlined for working with OpenShift bypass OpenShift's own web interface, for the most part, which are designed to help users get up and running with more conventional web programming environments like Rails and node.js. Of course, many developers might prefer to use those kinds of tools, but SignScape can help programmers get started if they are interested in a C++ and Qt-oriented foundation. Qt and C++ can be especially useful for projects building specialized front-ends as well as servers; particularly so in the science context, where there is extra need for sophisticated data visualization and numerical analytics, where C++ excels. Another benefit of

C++ as a web development language is strong typing: C++ provides one of the most comprehensive type systems of any major programming language, the more so with each new update to the language standard.

For web development with a science and technology focus, there should be extra motivation for choosing development tools which resonate with the culture of scientific research, and are connected by threads (even if only tangentially) to important current scientific and mathematical themes. Mathematical type theory is emerging as an important subject in foundations of mathematics, in the Homotopy Type Theory and Univalent Foundations programmes and the Curry-Howard Isomorphism. In a spirit of respect for these developments, programmers should embrace computer languages which recognize complete type systems, as close as possible to the far point of the "Barendregt Lambda Cube". For many programmers, this implies functional languages like Haskell, but there is room for sophisticated type analysis also to be recognized in the context of Object-Oriented languages like C++ or the Common Lisp Object System. While C++ will not in general express constructs like Dependent Types directly, static analysis tools can identify type criteria above and beyond the C++ compiler proper, so a new generation of type-analytic tools can bring ever rich type theories to the C++ environment. From this perspective, C++ is arguably as strong a language as any because of the quality of its development tools, with IDEs such Qt Creator and the opportunity to build plugins enforcing special extensions to the C++ language (such as, for Qt, the signal/slots notation and Meta-Object Compiler). This may be another incentive for preferring C++ as a web development foundation.

Modern web sites do not only serve web pages: they are multifaceted data management projects, curating collections of data which often should be automatically assimilated from external sources and shared with human visitors and computer tools directly. In a scientific context, sharing raw data is often an important form of outreach and collaboration. In a political and activist context, data aggregations helps invidivuals break the impasse of not finding events or organizations which promote the causes that most concern them. In a high-tech activist future, people will register with many groups that interest them and use aggrator software to unify information about events, news, policy recommendations, and books or other text resources. Curating information is therefore essential to both the scientific and political arenas, and should be that much greater a priority in a milieu which combines them. Strongly typed data models can serve as the backbone for online data management that both sends and receives HTTP content according to fine-grained data classification schemes, enabling organizations to develop and participate in APIs, data feeds, Publish/Subscribe architectures, and other collaborative data sharing models. These possibilities are further discussed in other pages here.

3 Data Management

The pressure and excitement to get a Web Site up and running can make Data Management seem like an afterthought. For many not-too-complex sites, Data Management can be informal, essentially a process of human editors making sure to keep the site up-to-date. Individual web pages, when they present information that is subject to frequent updates (like event listings), can be modified periodically as needed. A database-backed system, by contrast, will usually provide web page "templates" and automatically complete pages by inserting content from the database. This provides a guarantee that pages are up-to-date, since they do not rely on a person manually revising them. Many web sites do not use this kind of solution, however, because of the time, expense, and expertise involved in setting it up properly. Investment in a Data Management solution (whether in time or money) may not seem worthwhile, even if it can save time eventually.

For community-oriented, non-commercial organizations, there may not be a pressing operational need for sophisticated information management, but it can still pay dividends to incorporate such solutions, at least gradually. If these sites work well just with human editing, automation can be phased in, identifying tasks which people find tedious, or which often seem to be delayed or to cause operational difficulties. If a site is developed with "self-contained" (for example Container-as-a-Service) hosting frameworks, as discussed on the Web Hosting page, it is easy to create a "copy" site for testing and development, where automated functionality can be tried out. Creating and using management code also can often be done remotely, so volunteers work on their specific contributions on their own time and place. Organizations should decide whether such efforts are the best allocation of volunteers' time, but building a Data Management foundation, particularly one which evolves gradually and reflects concrete operational needs, can make future efforts more time-efficient. For example, while manually editing a frequently updated Event Listing may involve only (say) one or two hours per week, this is still an operational bottleneck, maybe without a good fallback when things go wrong, like someone's illness or vacation. The goal of a "Progressive Technology Network" is to help organizations share solutions so that more sophisticated capabilities, like automating certain Data Management tasks, can be adopted more easily, in a spirit of sharing and reciprocity across groups.

So assume a community of organizations which are committed to building and sharing robust Data Management solutions. The next question is what to prioritize when beginning to implement these solutions. In the internet age, technologists seem to have a reductionistic view on data management. In principle, data management is more than just maintaining a (perhaps web-searchable) database; it is more than just charts and visual "analytics"; it is more than data mining and "big data"; and more than data networking via XML or JSON. Instead, information has to pass through multiple phases of representation serving different computational environments, using tools suited for needs both of those directly responsible for producing data sets and those using them. Data Management tools and strategies therefore demand solutions which adapt to different phases of information "lifetimes", and different aspects of data representation.

Before mentioning a few technical points in term of data modeling, it's worth considering how this topic is important for progressive ideas and organizations. One factor is the challenges of distribution, as compared to production: even if the material hurdles to overcoming scarcity in a society are overcome, and the political hurdles to overcoming inter-group tensions which can undermine the political will to overcome scarcity in the first place, there remain the logistical hurdles of moving things from regions of plenty to regions of want. This takes many forms — transferring energy from wind farms and solar collectors to population centers, as much as shipping food from farms to cities, or piping clean water from reservoirs to households. In some scenarios, these problems can be given purely quantitative forms which make them directly amenable to software engineering — smart grids, for example, which can fundamentally redesign energy policy. Owners of solar panels can sell surplus electricity to utilities, making them more cost-effective; appliances can be programmed to operate more when power is more available. Energy distribution is modeled as a spacetime field whose shape has a criteria of optimization, with paths toward better states realized by sharing data across decentralized networks. With this technology well-understood for electrons, it is not hard to imagine "problem isomorphs", or near-isomorphs, pertaining to other civic priorities. The computations of distributing energy from source to sink is not that different from distributing products from factory to store — or, in a more civic vein, distributing food among soup kitchens, or books among schools. From smart grids to smart pantries.

This is not just a matter of calculations. A smart grid in theory is almost a purely mathematical object, but whenever data sharing and interaction comes into play, "computational" systems take on a qualitative as well as quantitative dimension. Consider networked appliances who modulate their

energy usage: to the degree that people have any control over this, they need a User Interface which presents the information they need to know — the scarcity of plentitude of power at different times, for example — and gives them control over functionality, like choosing when to schedule (say) the washing machine to do a load. There must be screens, displays, touchpads, and buttons, somewhere or other (on the machine itself or maybe on a user's computer or tablet or phone). And the "grid" data — in this example, about when electricity is less expensive — has to be shared from the grid to the device, to the computer or whatever the end user uses. A consistent data model has to be programmed into a multitude of network points, and realized in a visual and interactive front-end. This is orthogonal to the mathematics of the smart grid's "smartness" itself, no matter how sophisticated. The modern concept of a "data network" goes well beyond just a network of *computers*, finding its largest expression in the "Internet of Things" which not only networks devices but connects them to the federation of networks constituted by "the" internet. As much as it has great potential, this networked accessibility can go terribly wrong if data and capabilities are not carefully modeled and enforced: significant cyber attacks have already been carried out by exploiting the HTTP capabilities of networked devices. Perhaps "the" internet as a monolithic whole is too coarse-grained to ensure security in the realm of devices as well as computers, and within or parallel to the World Wide Web there need to emerge more targeted networks with more rigorous data models and protocols.

This is worth emphasizing because we need to guard against reifying the purely quantitative dimensions of computing power — "big data", analytics, and "smarts" is just part of the engineering problems standing between us and a fairer, more sustainable technology-driven society. It is fair to say that the "qualitative" dimensions of computer science are less clearly understood and researched than the quantitative: building data models which are consistent and reusable across a network, and building User Interfaces to make the technology convenient for people, as compared to just doing more "analytics". Code reuse is not only more efficient, because there is no time wasted creating near-copies of existing code, but also offers a tighter guarantee of distributed consistency. If source code itself cannot be directly reused — with different network points using different programming languages, for example — then at least formal data models can be reused, which can be seen as "code" in a sense — perhaps literally the code of a type-definition language, which can be compiled to and/or compared against executable code. In short, data management, particularly in a civic and/or "sustainability"-related context (one where negative environmental impacts should be mitigated, and positive ones reinforced), importantly includes the aspects of *reuse* and *front-end*: *reuse* of data models to ensure consistency; *front-ends* for User Interaction. Tools to design, implement, and when possible automate these dimensions, are as intrinsic to computer science as algorithms and data warehousing.

These are also dimensions especially suited to grass-roots, community-driven technological progress. Neither government nor large companies have been especially adept at the "reuse/front-end" aspects of information management, leaving deep progress in this area open to individuals, civic groups, and Open-Source projects. So "reuse/front-end" libraries can be shared among progressive groups to make their logistical operations easier and less time-consuming; and they can also be used to improve the general public's access to government services and their interactions with organizations. Not-for-profit groups can build alternative front ends to unwieldy government web sites, for example, or other entry points to government bureaucracies. Partly anticipating this possibility, governments are, to their credit, increasingly providing APIs for important information, in domains like transportation, health care, and financial support — although pressure has to be exerted on governments to provide more APIs and improve their "discovery". Assuming local tech groups can seize these opportunities, we can envision a technology-driven society gradually converging on certain norms of efficiency and accessibility: it will become less time-consuming for civilians to access government services, and for government and not-for-profit employees to management their various bureaucratic and organizational tasks. Moreover, the

momentum for this evolution may lie largely with civic groups that can shape the technology toward the common good. This optimistic scenario may not always come to pass, but it is certainly feasible in light of both progress and inertia in current science and technology.

What this means, importantly, is that Data *Management* — not just *analytics*, or numerical processing, but the "reuse/front-end" dimensions — points toward an increase in societal productivity overall comparable to those which in previous generations were driven by material or cultural changes. The 20th century saw societies become more productive because of better machines, better ways to generate power, attitudinal and workplace shifts, public eduction, the end of many forms of discrimination — a larger, more diverse, more educated supply of workers. The bounty of such an "increase in productivity" was not fairly shared, perhaps, but for better and worse it marked the societies and economies of the First World. What has changed in the last few years is the emergence of a new route to greater productivity — a greater logistic and operational efficacy driven by data modeling and software design — and seeing how the productivity of the last century was exploited by a few and led to widening, not narrowing inequality, our current chance of "soft" (as in software-driven, "reuse/front-end") productivity increase is also a chance at a do-over. For a variety of structural and technical reasons, "soft" productivity may be less readily appropriated, more readily shared, than the "hard" productivity of the past.

The science of data management, in summary, is the science of "soft productivity": its payoff to society is time shaved off the operational chores of government and civil society. For this reason it is important to shore up the scientific foundations of data management and data modeling, including exploring semantic and type-theoretic criteria, notions of consistency and compatibility of data representations across computing environments, security and data integrity checks, and tools for building the tools that make data management easier. This then brings up again the need for identifying different "phases" of the "lifetime" of data, and targeting solutions for each one.

There are at least five different stages or aspects of data which engender their own operational and representation norms: Database: Information expressed and encoded in structures suitable for long-term persistence Serialization: Information expressed and encoded in structures suitable for textual or binary transfer between different computing environments Typed Values: Information modeled and, during stretches of time, represented in live memory, as values which are instances of clearly defined data types. Document Representation: Information woven into human-readable documents like web pages or PDF documents. User Interface Representation: Data expressed as typed values presented as models to be visualized in UI components, including those expressly designed around the specific types, in terms of their layout, interactivity, and protocols for communication with other components.

Of these, well-designed data types are arguably the most important step. Any project which involves data management should have a library of source files representing the relevant kinds of data (e.g., C++ or Java classes). From that foundation additional types can then be developed to address the concerns of serialization, persistence, documents, and User Interface.

If *tau* is a type in its most direct (live) representation, the aforementioned alternative representations are "morphisms" which send *tau* to other types. The database representation of *tau*, for example, will often be simplified and condensed (for example, replacing pointers to related data structures with foreign keys). Serialization of *tau* instances, such as XML code, will be character strings whose range of allowable states is narrowed to conform with *tau*'s structure. In an Object-Oriented context, classes modeling *tau* can be provided with methods specifically for inserting *tau* instances into templates, including (say) XML templates for serialization, and (say) HTML templates for exposing this information in human-readable text (although classes should also provide an interface to access indiviual

fields to be used in text templates which the classes do not directly control). Finally, GUI classes are "Visual Objects" (for example, subclasses of QWidget or QDialog), and for any type *tau* there is potentially a variant of a QDialog, for example, whose specific purpose is to show and edit *tau* instances. Data Management code frameworks therefore need to help programmers create the relevant libraries and source files to cover these various scenarios organized around underlying types.

Given these requirements, Qt in particular is one of the leading resources, widely considered the best tool for cross-platform Application development. Qt, in particular, supports User Interface development which sidesteps some of the more obscure programming details of frameworks like Microsoft Foundations Classes, but also provides user-friendly front-ends with Native Look and Feel that clearly outperform languages like Java and most scripting languages — except insofar as they wrap Qt classes themselves (as with PyQt).

Native-compiled, cross-platform components to show, edit, and interact with individual type instances — as well as their lists and collections — can make data curation much easier, for those preparing data sets for web and data sharing purposes. Potentially, these visual tools can also be useful for those consuming data, although it is more complex to design native software to run on end-user's computers, compared with providing a web interface. Those building web sites which involve data sharing can decide whether it is worth providing custom software for end-users specifically designed to work with their resources in particular. In other situations, a decentralized group of individuals or organizations seeks to curate data collectively, so they may share custom software amongst each other even if not with public consumers; this allows more leeway for the software to go through a period of development and refinement. In non-commercial setting, software can be designed in consultation with a specific user group, and can evolve organically. Off-the-shelf commercially-produced software may be expertly designed, but it is also targeted to a broad user base with varying needs. Custom-built applications priotize the capabilities most important to its specific users and make the most common functionality the most readily accessible. Custom software development is also a chance for progressive organizations to support community-based and non-profit computer programming. For these reasons, data management via custom software tools can be an important project for many scientific and/or progressive communities.

4 Data Sharing

Data Sharing is very important for scientific and activist communities alike. For science, a spirit of collaboration means that scientists are not only sharing ideas and papers, but often sharing raw data directly. In politics, groups united by similar causes and concerns need to work in consort. Events and news stories which are of interest to members of one group will often be of equal interest to members of other, loosely affiliated groups. By designing resources in a fashion that they can be readily integrated, the information published in one place gains in value as it merges with related information available elsewhere. The goal of data sharing among progressive organizations is to curate data collectively into a rich platform which is greater than the sum of its parts.

Moreover, software tools can help governments provide important information to members of their community, and vice versa — we should expect government transparency, user-friendly forms and access to information, and streamlined, effective access to government services and resources. We should not however expect politicians to embrace these principles by default, however, in the face of budgetary or even ideological restrictions and bureaucratic inertia. Activists can take the lead partly by advocating for a more streamlined, computationally sophisticated paradigm for government/community

interaction and partly by actually building those tools — whether they are directly used and endorsed by governments or independently developed and maintained to supplement official platforms (custom front-ends to government APIs, for example, or guides to using government services). Progressives' ingenuity, and (often) academic millieu, can spearhead grass-roots solutions to complex contemporary problems in "Community Informatics", "Sustainability Design", and related Public Computing disciplines.

Many areas of science and medicine are increasingly finding a need for heterogeneous data representations to be integrated. Vast amounts of information are housed in traditional Relational Databases, along with more recent NoSQL models like "Column" and "Key-Value" stores, and graph-oriented (including Semantic Web) information spaces. To implement querying and analytics crossing these paradigm boundaries is a major technical challenge, and limitations in existing technology's adaptation to "heterogeneity" problems have apparently contributed to numerous high-profile glitches, including the botched roll-out of the US Affordable Care Act. Such technical issues are particularly evident in Bioinformatics and medicine, where (spurred by an intense push for Electronic Medical Records and related cross-institutional data sharing) there has been significant investment in "Semantic" technologies as well as digitizing and integration of existing data sources and data-collection tactics (gathering patient and community-health information, for example). Since Health Care — and the interrelationship between governments, health providers, and communities — are important policy, justice, and governance issues, this is an area where Progressive and scientific organizations should share interest and engagement. It serves the general public as well as governments and institutions to develop tools for creating and receiving information which are more robust in the face of structural incompatability between disparate data sources.

Along these lines, "Data Integration" problems are certainly well-known to both academic and commercial communities, and much advanced research and technology has been developed in response. Scientific and Progressive interest again align, however, in recognizing that a measure of solutions' successes should be cost, reproducibility, availability — an expensive or limited-access technology, which provides innovation and improvement for a select group but is not designed for large-scale use and dissemination, or contributions from an open, non-profit-driven community — does not necessarily have equal scientific merit to one which serves a broad public. This is a point worth emphasizing in a world where "Big Data" analytics is recognized by many as having "game changing" potential, but reifying technological progress can blind us to how greater capability also often means greater inequality. If a given technological platform allows unprecedented efficiency, knowledge discovery, product development, etc., but is available only to a few select groups or individuals, whatever conceptual breakthroughs it may embody can hardly be called true "progress". This is not only an ideological concern, starting from an assumption of common utility, but a basic scientific concern, starting from a desire for scientific rigor.

A technology whose availability is narrow and privileged is less sophisticated than comparable solutions whose benefits are more broadly replicable. In Computer Science, the qualitative aspects of software and programming — formal language design, User Interface design, code generation and analysis, Semantic and type modeling — are equally hard, arguably harder, as more directly quantitative problems. Quantitative methods have evolved to a point where impressive results can be achieved with sufficient computing resources, but this is still a kind of progress which favors the already-powerful; it is instructive to see how many language-design and other "qualitative engineering" problems remain stubbornly resistant, for all the money and attention given to phenomena such as "Big Data". Consider, for example, how many years it takes for language designers to implement recommended features for successive language standards, or the fact that programming language choice is often a question of what limitations in the language are least bothersome — it may be impossible to do serious program-

ming in any one language without encountering desired capabilities that are present in other languages but have to be at best approximated once a project is underway. These examples illustrate that certain high-profile capabilities of "Big Data" and quantitative methods — decent machine translation, self-driving cars, Optical Character Recognition (to make depositing checks more convenient, for example) — should not be construed as evidence against the power of community-driven, non-commercial technological problem-solving, which can have greater scientific as well as ethical merit. Tecnology solutions which work via APIs, Open-Source components, and modular design, where collaborative projects can extend and disseminate various parts of the code base, are more valuable to the public than are tightly inter-connected and self-contained solutions, no matter how advanced. It is important for there to be technologies which addresses issues like data incongruity — and promotes emerging solutions like the Semantic Web — and are also committed to serving the common good openly and non-commercially.

This goal has ramifications for how data sharing platforms are designed. Generic data sharing includes several variations, such as APIs, Data Feeds, and Publish/Subscribe systems, which differ in terms of the degree of interaction between the data source and its clients. In a Data Feed, the data source makes some information statically available, like a list of upcoming events which automatically updates when a new event is added. An API, by contrast, will respond with more focused information when given individual queries — for example, providing information about an event when given the date on which it is scheduled. A Publish/Subscribe system is like a Data Feed which maintains a list of clients to be notified when information changes. For example, emails can be sent out when a new event is added to a schedule. These systems can work in consort: a Publish/Subscribe system connected to someone else's API would be an example of Publish/Subscribe with a more rigorous data model. Instead of sending emails, the system would send messages containing structured data indicating that an event has been added; software would subscribe to receive those messages and react to them by updating its own internal database, which in turn would be queried through an API. Despite this range of systems, data sharing in general is often discussed through the lens of API design, so a valuable contribution can be both tools and strategies for Progressive Organizations building APIs.

In light of "heterogeneity" issues as mentioned, one challenge for API design is anticipating inclusion in networks whose data sources reveal a spectrum of distinct data models. In response to this, Computer Scientists have tried (especially over the last decade or so) to envision a "most general" representation framework, one which can incorporate Relational and various NoSQL data models without being constrained by them. This was an impetus for the Semantic Web and formats like RDF, as well as other graph-oriented models like Conceptual Graph Semantics, descended (at some remove) from Peircian Epistemology. Other "Semantic Data" perspectives also reflect philosophical backgrounds, including Phenomenology and Philosophical Ontology. But despite this deep pedigree, efforts like the Semantic Web have had only limited success, failing to influence web and application design on a large scale (even if related technologies are becoming indispensable for a few large-scale and high-profit branches of industry, especially in Health Care).

A corollary issue is therefore how to incorporate Semantic Web data and/or similar models in a more "grass roots" network of software. One option worth exploring is combining "Semantic" Data with Open-Source code, on the premise that freely accessible code libraries can themselves serve a Semantic layer or specificity for data networks. Serialized data structures, for example, could refer to Open-Sources libraries as annotations defining type structures, which can then be attributed to serialized values. A serialization, according to this proposal, would include a collection of named types mapped to web addresses for code libraries, which then provide Semantic grounding for attributions like rdf:type. This differs from Semantic "Ontologies", XML Document Type Declarations, and RDF

type specifications because each of these alternatives attempt to provide a "universal" type model abstracted from any specific programming environment, which (while perhaps well-intentioned) often makes data sharing harder, not easier. Since information must originate and culminate in application code, "abstract" type models still need to be realized in concrete implementations; and despite the Semantic goals of achieving a truly universal and "language-neutral" representation, it is hard for such "generic" formats to be truly independent of developers' preferences influenced by their preferred programming languages. In other words, the goal of "universal" type models appears to be subject to inner divisions and spinoffs, which defeats the purpose — not one but multiple "universal" models, reflecting the emphases of different programming language communities. If this dependence of universal models on concrete languages is unavoidable, perhaps a reasonable solution is just allow languages themselves to provide canonical type models, which can then be adapted for different languages as an example of regular "porting" of libraries between languages.

Notwithstanding the vision of "Universal Modeling", there are a significant range of data structures that are better represented in terms of conventional Type Theory rather than Semantic Web "Ontologies" and other "Semantic" constructs. The Semantic Web is, first and foremost, conceived in terms of binary object-to-object relations, which is an important genre of assertion but hardly comprehensive. Such binary relations do not directly capture data "collections", like ordered and/or unordered lists, queues, or stacks — note that queues and stacks differ from lists in explicitly defining the norms of their change over time: queues are "First In/First Out" and stacks are "First In/Last Out". Semantic "triples" are generalized to relations of arbitrary arity within Conceptual Graph Semantics, (CGS) which also recognize relation hierarchies, with the result that relation "signatures" end up being formally quite similar to type-theoretic functional signatures. But CGS still does not straightforwardly model "Collections" structures. Or, to be more precise, both "triples" and CGS can *represent* arbitrary data structures, but we need to distinguish between representing *encodings* of data structures from representing the structures themselves. Semantic Web structures are indeed a powerful framework for representing *formal languages*, including languages that textually serialize data structures of any kind, and in this sense the Semantic Web indirectly does provide "universal models"; but this is less a matter of a universal *semantics* as a globally applicable *syntax* to which various formal *serialization* languages — encoding different forms of data types and data structures — can be converted. This universal representability should not be miscontrued as a Universal Semantics.

Instead, Semantic Web representations and Ontologies are most useful in conjunction with formal types and data structures, not as their replacement. It is therefore useful to assume that, along with any notion of "Semantic Web", there is a universe of Typed Values (the emerging fielf of Category-Theoretic Database Analysis provides an interesting mathematical foundation for such type/Semantic integration). While Semantic Web structures can be used to *serialize* Typed Values, this (as just argued) is more about syntax than semantics, so we may assume that the specifically *semantic* contribution of Semantic Web is to *assert* relations *between* Typed Values that are not directly modeled within type-specific semantics. In this sense Semantic Web can usefully expand the range of a type-oriented Semantics by providing models for information which can be formally represented in terms of types and values but is not found within types themselves.

The Semantic Web — in this proposal — provides a space of "assertions" which have Type-Theoretically defined "signatures", but rather than representing fields or functions defined specifically *within* types, assertions are *outside* type definitions and orthogonal to formal type systems, in the sense that an assertion is not itself a "typed value". Assertions do not "have" types, though they have signatures which involve (two or more) types. Assertions can therefore capture inter-type relations, including those that are not accounted for in each type's defintion. This includes relations between

typed values, but also relations between types that hold for all their values; for example, the fact that a field in one type represents an inverse relation relative to a field in a related type, such as the relation between an author and a book (which presumably represents a field present in both an "author" type and a "book" type, but the inter-type assertion would be that the two fields are interdependent).

So, in sum, the Semantic Web provides a useful extension to ordinary type systems so long as its structure supplement, rather than attempt to replace, those of the underlying type systems. Meanwhile, any reasonably sophisticated type system needs some notion of *term algebra* and therefore some mathematical model of formal language (to define well-formed terms, which are prerequisite for expressing any term-dependent type). Moreover, the expressiveness of "formal language" in modern computing environment requires more complex structures than lambda-calculus or even first-order-logic notions of term algebra; Semantic Web representations can be a rigorous alternative. So Type Theory for modern programming languages may indeed rest on some notion of Semantic Web representation, alongside the fact that Semantic Web depends (to be properly Semantic, as argued here) on modern Type Theory: the differences is that the former dependence is mostly *syntactic*. In other words, "the" Semantic Web really has two different dimensions, one Semantic and one"syntactic": as *syntactic* the "Web" forms a variation on term algebra upon which richly expressive type systems can be defined, and then on that basis the *semantic* dimension of the Semantic Web comes into play as a space of "assertions" extending the space of information which is represented by Type-Theoretic structures by themselves.

Another question for API design is the preferred format for encoding and serializing data — that is, for writing it as a text stream that can be sent over a network. The most popular formats are mainstays of web development like JSON and XML; another class of formats (sometimes embedded in XML) are Semantic Web languages like RDF and RDFa. More exotic formats are derived from Lisp, such as EDN (Extensible Data Notation); the main advantage of these encodings are "homoiconicity", meaning that identical notation is used for representing computer code as for representing data. Semantic Web representation is popular among some specialists because modeling Semantic Web data involves not only choosing a formal language, but designing data models (often published as "Ontologies") which represent semantic and conceptual details. In other words, providing Semantic Web resources commits projects curating data to think carefully about the semantic meanings of information — the concepts which need to be modeled to make sense of the data, the properties or observations which must or may be associated with distinct entities, and logical relationships (along the lines that every employee by definition works for a company, any author has written a book or text of some sort, and so forth). While this modeling is important, it is not necessarily true that these logical and conceptual considerations should be intrinsic to data serialization. A data sharing model based on strong type systems has many of the advantages of Semantic Web but is arguably more pragmatic from a software engineering point of viewer, with a clearer "Separation of Concerns".

On the face of it, the only responsibility of an API is to provide data, usually by having a capability to respond to HTTP requests with structured responses in a format like XML. On the other hand, to make APIs easier for others to use, API providers should seriously consider providing source code for API clients, rather than relying on external programmers to create clients based only on API documentation. Developers still have the choice of building separate clients or incorporating the provided libraries into other projects, but initial client libraries both offer a starting point for this further coding and they help document and explain the API. This design — where developing an API includes developing both server and client code and treating the API as a means of passing data between them — also influences how the API itself is conceptualized. The APIs design in this sense is successful if data is consistently reconstructed on the client end that is compatible with the original (server) data. Assuming a strongly typed environment, this means that the data type for values reconstructed on the

client is either an identical implementation to that on the server or is a "compatible" type, for some suitable notion of compatibility. What constitutes a proper data serialization is relative to the needs of the functions which map the received data to the client data types — XML serializations do not need to be verified against a Document Type Declaration, for example, if the client code can manage potential missing data or if the server code can be guaranteed to provide essential data.

Integrity checks like DTDs, as well as Semantic Web models like Ontologies, attempt to place restrictions on representations themselves in the serialized format. In terms of Separation of Concerns, however, the content in this format is only one representation of data, serving a particular purpose (the capacity to be sent over a network, usually in a textual format). Integrity and Semantic concerns really apply to the typed values which produce the representation and which receive and reconstruct them, rather than to the representations themselves. This is why focusing on designing strong type models for the data in its server and client state is arguably a more rigorous application of Semantic Web principles, and encourages a theoretical unification of Semantic Web with formal type theory. It also implies that there should be a converge of Data Sharing with Open-Source code hosting, insofar as open code libraries would serve a more intrinsic role anchoring the Semantic precision of data according to its intended interpretation and integration.

Giving these considerations, a case can be made for XML as the data format of choice: it is more robust than JSON, more narrowly focused than Semantic Web formats, and has an extensive inventory of tools for almost any programming language. XQuery, in particular, is a powerful tool for extracting data from XML serializations and using it to populate instancs of newly constructed typed values. ScignScape employs a data sharing protocol based on XML which uses XQuery to generate Clasp code, which in turn then relies on Qt bindings to reconstruct datatype instances from XML. This protocol depends on the Qt Meta-Object System (MOC), which is a powerful tool for C++ reflection. Cumulatively, the combination of Qt's XQuery and XML patterns modules, Clasp, and the MOC are a good foundation for designing API client libraries, and with Docker supporting server-side Qt there can be considerable code overlap between server and client side. This is only one of many options for designing integrated API solutions, but it is a good example of API design in native-compiled and strongly-typed contexts.

Tools which help organizations develop data sharing systems like APIs can streamline other operations as well, including preparing web pages to begin with, and also adding capabilities to web sites for purposes like Search Engine Optimization and consistent User Navigation. Site maps, "alt" tags for images, and other details which are recommended for professional web design (but often omitted from less technical web sites) provide benefits both for SEO and for site visitors. As groups make an effort to keep web sites organized and up-to-date, this work can be extended fairly organically to maintaining APIs and data feeds. If a collection of "Progressive APIs" can be built up, across multiple organizations and unifying different aspects of politics and activism, it would be possible to develop end-user tools for engaged individuals, as well as operational tools for organizations and their employees or volunteers. Imagine software or an "app" which scans multiple APIs for upcoming events and notifies activists of things they may be interested in, prioritized according to the causes they support the most (alongside obvious criteria like geography). Certainly Social Media has been a valuable tool helping activist communities to coalesce, but we should be wary of assuming that the interests of Social Media companies align consistently with Progressive values. Furthermore, "activist software" can be published as Open-Source to allow for collective customization and refinement — developed on native platforms, for example, which can make the software more usable than generic web applications.

This site will gladly link to APIs which essentially fit the "Progressive Technology" theme. I would

also be eager to help organizations interested in implementing advanced data sharing systems using some of the above principles (with Qt, Clasp, and XML); the code on GitHub provides a starting point, and can be readily customized for specific purposes, for example building type-specific GUI clients for APIs. Please visit the Contact page for more information.

5 Documents

This page provides a collection of links or references to papers which touch on themes related to science, humanities, language, society, and politics, and their intersection. Works like these reveal a truth about knowledge and our intellectual landscape in the 21st Century: our inherited picture of the partition of inquiry, between Natural Science and Humanities for example, is eroding. Many of the most fertile lines of research — in commerce and industry as well as academia — involve both a scientific and a humanistic component (taking these both broadly). Natural Language Processing, Artificial Intelligence, Educational Technology, Computer Graphics and Computer Generated Imagery, Virtual Reality, Human-Computer Interaction, Data Management, Robotics. An intense focus of scientific research is making sense of, facilitating, and in some degree recreating specifically human patterns of thought and language, and vision, experience, and embodied, enactive pragmatics. Instead of science *or* humanities, there is now a third region of academic which depends equally on both.

We should not be swayed to accept a reductionistic view of human nature, as if industry can create human-like robots and replace human agency, populating our roads with driverless cars, using software in place of humans to communicate with children in school or patients in hospitals, relegating ethical choices to computer programs, and so forth. There is something dehumanizing in assuming human reason and conscience can be emulated so facilely. But this should not compel us to ignore the serious science of Artificial Intelligence and related disciplines; on the contrary, we need critical engagement with these dimensions of science lest they be appropriated by a "Computational Industrial Complex" seeking quick profits and wielding considerable influence by monopolizing our technologies for sharing ideass. The new science/humanities overlap — which perhaps can be called "Formal Human Sciences" or "Computational Human Sciences" — at its best gives us rigorous pictures of certain cognitive and structural patterns which are intrinsic to our use of language and our engaged perception of our environing world, while still recognizing a further dimension of knowledge and consciousness which is embodied, intersubjective, experience-based, and situational to the extent that formal simplifications lose something essential. Formal models of thought and cognition are deliberate simplifications, which may be scientifically truthful for that very reason while also failing to capture the nuance and essence of human nature. We should accept their scientific merit so long as we always keep in mind their existential limitations. For this reason, a considerable range of Cognitive Scientific and related literature has a place in humanistic and even political research projects. It is important to develop document collections, and more sophisticated tools as well, for cataloguing and extracting patterns from the mine of texts and investigations that shed light on this overall science/humanities overlap.

In addition to linking external documents, this page can provide specially curated papers designed for more rigorous data sharing based on structured information embedded and referenced within documents. This includes papers prepared with markup sources that can be preprocessed with software tools to extract data like lists of authors, web links, and data sets or graphics which might be of interest used in other resources. Several annotated papers are listed below, with links to the articles themselves (in PDF format) paired with links to automatically generated web pages derived from annotations inside the documents. These accompanying pages show lists of names, certain technical subjects (like computer technologies and programming languages), and links to referenced articles. They also have

direct representations of the markup sources (converted to Khi Forms, in particular) just as an example of how publishing sources can facilitate a collaborative design of programs that analyze and extract data from documents. SciGnScape provides a Lisp library which reads Khi Forms and can be supplemented with functions that process the text looking for specific kinds of information.

6 Defending Science

No doubt, science is under attack from forces around the world which would undermine Democracy: the skeptical, deliberative ethos of science at its best is evidently threatening to people who want to perpetuate social norms and power structures without popular consent or rational scrutiny. Science is also, more subtly, under attack within Democracies themselves, because the policies which reflect scientific consensus do not always coincide with ones which many prefer for social or economic reasons. Climate change is the most obvious area for this kind of controversy, although there are other voices showing how environmentally sensitive entrepreneurship can also be socially sustainable and economically viable. In either case, attacking science — like attacking journalists, education, and minorities — is a recurring trope of political movements led by authoritarian ideologies and personalities. And yet pro-democracy movements have not always responded by vigorously endorsing and defending science. The "Postmodern" element of the Academic Left is even considered to be hostile to science in some sense, at least relative to superficial analysis of foundational Epistemological concepts like objectivity and scientific method.

It is encumbant on scientists — and scholars in general who are concerned with science — to not only *be* scientists and researchers, but to *defend* scientists and researchers. To be a scientist should be an *identity*, expressed differently in different people for sure, but aware that this identity is sometimes and in some places subversive, outsider, controversial. To be a scientist is to be an agent of change. The politicization of science may flow naturally from a cultivation of scientific identity, making this part of the discourse of Philosophy of Science and understanding of what "doing" science concretely means. Scientists — including human and social science also — should not hesitate to acknowledge that the very fact of *being* scientists (in acedemia or elsewhere) implicitly positions us in a political matrix; it includes in our personal identity a relation to Creation, State, and community that is politicized and potentially controversial. To recognize this is to build a fellowship, even subconsciously: we see in the social and political travails of other scientists and scholars, around the world, a reflection on and threat to our own situation. We become politicized not only through the cerebral form of deliberative ideals but through the emotional resonance of concern for others whose lives, in a plausible parrallel world, could be ours. A scientist or journalist who reads of a fellow scholar's or writer's imprisonment instinctively responds as does the transgender individual to a hate crime, the black American to racialized police brutality. The politics of science is an identity politics.

Identity politics does not have to set out as a political movement; organization and structuration emerges gradually, as people translate inner feelings to solidarity and desire for change. When science migrates from theory to practice — from building technologies to applying them for social value — the political vulnerability of science becomes apparent, because to imagine science and technology in practice is to imagine community leaders trusting in science and reason implementing rational policies in the context of a larger society, with an often everpresent threat of resistance from entrenched interests. The theory-to-practice transition may not be politicized from the outset, but in the global situation as a whole it will become politicized before long. And yet as science becomes politicized it can do so on its own terms, applying its intellectual strategies to its own political needs: to share information about science's political standing, the legislations pro or counter to scientific interests and consensus, the so-

cial position of individual scientists and academics around the world, the needs of activist communities resting their policies on science and technology — to share and track this information is a challenge of knowledge management, information engineering, software design. Curating pro-scientific activist data is already a political act with potential tangible payoffs, to know which policies or which international figures to write about to which political leaders, to find talented scholars for whom visas and scholarships can be a path to realizing their social mission or even freedom from oppression, to know what kind of contributions and in-kind support would benefit pro-Democracy organizations that the academic/scientific community is particularly able to provide. Imagine self-consciously pro-scientific analogs to Amnesty International; Github; Crowdsourcing. To build a digital ecosystem of scientific politics is itself a scientific problem, and demands scientific solutions.

It is certainly true that science has not always lived up to its best ideals; too often science has allowed itself to be manipulated for profit or ideology. And the flip side of scientific method — with its appreciation for empirical evidence and rational justification of theories — is a smugness that can suppress alternative voices. Those who are scientifically literate and embrace a scientifically sanctioned world-view can be too sure of their own rationality, their lack of prejudice, or the correctness of their beliefs. Perhaps indeed this is driven by distrust in intuition or cultural heritage which is exacerbated by scientific kinds of reason and science's evolution in a specific, "Western" culture. But science also has the potential to self-correct, challenging received wisdom from its own past no less than from elsewhere, maintaining an inner- as well as outer-directed skepticism, allowing outsiders to present both empirical and argumentational challenges to paradigms, pointing out flaws in reason or evidence or both. The defense against scientific self-satisfaction is not to reject science as an intellectual discipline, but to dismantle social hierarchies in scientific institutions: ensuring math, science, engineering, and technological education for all, encouraging underrepresented groups to pursue scientific careers, challenging the irrational accumulation of authority within scientific institutions, or the preeminence of a few schools and journals. Science must be willing to recognize talent and meritous theories in many places — irrespective of race, gender, national origin, or the polish of a persons' maneuvering through the social institutions of science, as opposed to the conceptual structures of theory.

Science which fails to live up to these ideals easily builds mistrust among the public at large, undermining its ability to shape policy and influence culture. The picture of science as cold and dehumanizing may reflect bad science, but scientific institutions bear some responsibility for this image when it holds sway. Research fields seem sometimes to contract a dangerously mechanistic world-view that holds sway for a generation or two before gradually fading into something more nuanced: psychology, medicine, and biology (particularly with respect to higher animals) have all gone through phases where empathy, subjectivity, and even moral valuations became disreputable. Animal research can be especially egregious in this regard. The risk of antropomorphizing animals (which is real, to some degree) does not warrant our ignoring animals' capacity for pain, happiness, emotion, social bonds, intelligence, or sense of self. While they lack our complex symbolic systems and our capacity for far-flung knowledge, the raw joy of contentment in social and physical environments which match their natural proclivities should no more be arbitrarily denied to animals than to humans. And since we cannot precisely imagine how a "good" life feels for a Bonobo or a bird, our experience of friendship, or of hunger satieted, may be elemental enough to project outward and offer a sense of what many animals do consciously experience. Or, switching to human medicine, we cannot ourselves experience others' states of physical or mental illness and their improvements, but first-person reports offer a window onto real states of affairs that should at the very least be considered alongside more clinical data. If alternative medicine makes someone feel better, for example, and since such improvement is a primary goal of medicine, that fact needs to be taken seriously — skeptical hesitations, such as suspicion that any improvement is mostly "placebo" and therefore may be only tempory, are not exactly irrelevant

(since after all the best case scenario goal of medicine should be not temporary healing, but permenent wellness) — but they are addendums to the tangible facts that patients do sometimes feel better. Again, givens which do not fit a theory are still givens unless there is compelling reason to deem them illusory — and as Philsophers of Mind have argued, the idea of "illusory" experiences is especially problematic in subjective realms like pain and depression or their absence, since (for example) phantom pain is still pain.

There is nothing intrinsic to science which mandates an irrational skepticism against subjective experience, or empathy or introspection — even if *some* sciences at *some* stage in their history have perceived a need (real or imagined) to treat these forms of reasoning with caution. Overcommitment to empiricism is no less antiscientific than is undercommitment. If animal researchers have at times felt institutional pressure not to identify too strongly with their subjects — not to "anthropomorphize" them or to be "hindered" by ethical concerns driven by tangible empathy for subjects' mental and physical pain — this has nothing to do with science and everything to do with the institutions. Each scholar is free to employ forms of argumentation which they find compelling and which further the structured pursuit of knowledge; if the kind of theory a researcher finds convincing — the kind she would appreciate if she were its reader and not its author — includes subjective and empathetic judgments, then it is her right to theorize as such and her responsibility to build a theoretical and metatheoretical scaffolding that includes subjectivity and empathy in an overarching rationality. This is not guaranteed to work, and subjective judgments may indeed lead to poor methdology — but the absence of subjectivity may do so just as well. Even in mathematics, attempts to completely eradicate subjective judgments may be bound to fail against Russell type like paradoxes (or, picking up an argument from Husserl's *Formal and Transcendental Logic*, against model-theory complications like L öwenheim-Skolem). Science needs intuition and subjective judgment, as it also needs checks against a collectively observed world. How each scholar finesses the balance between subject and object is a reflection of her own craft. Attempts to mandate a certain construction of this balance as theoretical preconditions for one subdiscipline or another are examples of paradigmatic over-reach: sure, some consistency in how the subjective-objective balance plays out helps characterize each field and preserves a measure of theoretical autonomy, but the consolidation of norms along this vector as along any other aspect of theory-paradigms is only legitimate if it happens organically, with the rational consent of all serious researchers devoted to the subject matter. In this sense the perpetuation of theoretical norms in a scientific field is quite analogous to the perpetuation of laws in a Democracy. For this reason any trace of class, race, gender, or any other prejudice in scientific institutions, or any deference to institutions on reputation rather than rigorous deliberation, is inherently unscientific — it undermines scientific legitimacy no less than doctoring research data or fabricating evidence. Each scientist's responsibility to shape theoretical parameters by the tenor of her writing and argumentation is equal to her data curation; the conceptual matrix where data is interpreted is as fundamental to science as is data itself. Inheritors to the early-modern Rationalism/Empiricism debate, we cannot ignore that both dimensions are essential to science, even if "Rationalism" in our time is understood first and foremost at the level of textual performance (manifesting patterns of argumentation in theoretical discource) — since texts are how ideas are disseminated in our time; the scholar is first and foremost a creator of texts — whereas in (for example) the early Royal Society scientific Rationalism belonged more to the order of scientists' private mental imaginings and to the social intimacy of lectures and live experiments. Since this intimacy has an elitist dimension, it is quite appropriate that scientific "Rationalism" orbits around scholars as producers of text, which is something that does not require expensive apparatus, or social standing. Anyone, rich or poor, white or black, male or female, can create successful scientific texts. And any successful text must be read and respected.

So, here the interests of science and of pro-Democracy movements coincide. Education is a right

— and that includes "STEM" fields (Science, Technology, Engineering, Mathematics, Medicine) and technical subject areas overall. STEM education lays the foundation for a technologically advanced and efficient Civil Society. Pro-Democracy movements should be encouraged to make universal STEM education a part of their platform, and we can support them by sharing or creating educational resources — translating texts to a wider range of foreign languages, for example, and refining pedagogy to consider local culture and everyday life. And this requies an interdisciplinary perspective spanning learning theory and humanities as well as mathematics and natural science. Considering the important overlap between science and humanities, an emphasis on STEM should not preclude equal emphasis on human and social science. Where a Democratic Civil Society is still a work in process, it is still more important to bring a technical mindset to humanistic applications: tools and software for local languages and dialects, management solutions for sociological data, Computer Aided Design for Civil Engineering, tools for education and field medicine, and so forth.

Scientists need to value applications of technical theories in civic and pro-Democracy settings as possessing considerable value, equally supported and promoted as "pure" research. Science needs a global perspective, rather than carrying on as an exclusive subculture living mostly in a few prosperous First World cities. Online publishing and related digital media have undermined some of the rationale for privileging a few standardbearer authorities. Yes, clearly articulated arguments are more likely to have a coherent internal structure that will hold up to future scrutiny; and instutionalizations of scientific authority — via peer review and academic scarcity (not everyone gets scholarships, degrees, accepted into prestigious departments) — provide a defense against science becoming careless and ideological; against "pseudoscience". But we should not assume that quality science (in any sense: natural science, social science, humanities) always bears the imprimateur of elite institutions. Structured reasoning signals responsibility and respect for deliberation through the caution of its presentation, the detail of its argumentative to and fro, and the precision of its terminological and theoretical tools. The imprimateurs of quality, responsible writing come from inside the text itself.

A new generation of online tools, journals, publishing platforms, and digital resources can provide the infrastructure for a more global, egalitarian scientific community. This is a technical challenge, not just a social principle: a wider range of papers and subjects should be given a forum, interdisciplinary research should be more eagerly promoted, and a range of voices, with greater diversity and internationalism, should be recognized and respected. But this demands better annotation and organization of documentary resources so that readers and scholars can find resources that interest them. These challenges show that a more contemporary and inclusive view of science does not imply a dilution of technical rigor: on the contrary, it is a more Democratic view of the global scientific community which embodies a more technologically sophisticated picture of how scientific resources are curated and evaluated. A scholarly paper carefully annotated, perhaps translated to multiple languages, integrated with linked data networks and presented within a software ecosystem which makes the text and its accompanied data highly interactive, all represent a level of technical achievement beyond the norms of today's academic publishing. In the future, the sophistication of the platform may equal peer review and academic reputation as guardians of academic rigor: an advanced technical foundation at the publishing and document curation level can support a more inclusive, egalitarian, and global research community.

The relation of science to society and politics is complex: on the one hand, science models a non-prejudiced rationality that is open to all voices; on the other hand, this very rationalism can lead scientific communities to reject too quickly ideas and values which may in the future be revealed to have scientific merit. People whose pain or discomfort, for one example, are helped by "alternative" medicine derived from non-Western culture, provide case-studies which science needs to study, not dismiss. The

traditional explanations for Alternative healing benefits may not be satisfying, but that's a challenge for a culturally informed science which takes patients' experiences seriously to take on, not a premise to ignore actual states of affairs. Or, for another example, traditional farming practices, which in past centuries were seen as backward and inconsistent with Western science, have more recently often been embraced as actuallt consistent with modern ecology. When science is applied to real-world endeavors where social and cultural factors are also in play, the humanistic study and appreciation of the cultures involved needs to be recognized as an intrinsic part of the science. On the one hand, scientific method is a model of Democratic openness and deliberation; on the other hand, the rigors of scientific method risk decaying into a presumptive elitism, a knee-jerk seduction by prestige and presentation — an over-investment in the theatrics of formal academic writing, with its many layers of signaling authors' qualifications, of their knowing the routines of writing for a technical audience. To work through these competing forces, it is important to attend carefully to the full picture of scientific reason and method, the similarities and differences across scientific fields and across the division between natural science and humanities, to approach notions of empiricism, argument structure, evidence and warrant, theory-building and the nature of theory languages, with considerable granularity. Oversimplistic pictures of science and its method can lead us back to the circle of contradictions which can undermine science's positive benefits to society. The politics of science, in other words, need to build on a rigorous Philosophy of Science, and to consider human and social sciences as on equal footing, from a meta-theoretic perspective, with natural science and mathematics. Argumentation in both broad areas needs to be rigorous examined, compared, and best practices identified and encouraged.

While surely philosophical and academic, I believe this analysis forms a precondition to intelligent engagement with science in a political context. Those working on the science/politics frontier should redouble our efforts to engage this kind of analysis; perhaps new journals or web resources are needed. Suggestions are welcome! This site can perhaps evolve into a forum for people and organizations that want to promote science in a social and political context: defending science against government repression, and promoting science education, like all education, as a fundamental human right.

7 Science, Politics, and Humanities

Any comprehensive vision for science must include and feature the Humanities — whether a political commitment to serve the common good through science and to model pluralism and Democracy through proper scientific method; or a Philosophy of Science to which is assigned the exploration and clarification of scientific method consistent with this political purpose. A philosophy must rigorously consider the perspectives and methodologies of human and social sciences, their similarities and differences among each other and in comparison to natural science, if it endeavors to realize science holistically. Fortunately, a facile contrast between science and humanities, as if the realms of knowledge and academia can be crisply divided into these two separate camps, is increasingly undermined by science and technology itself. No-one should deny that computer technology, data analytics, information management, and the engineering of modern civil societies are among the most fertile areas for 21st-century science — but they are also deeply intertwined with humanities and social science, needing sophistical models of langauge, thought, social relations, and perception, which can then be approximated or facilitated by technology. Computational tools may try to understand language or at least make people more productive when working with Natural Language resources, like researching and preparing documents; they may try to emulate human vision or at least employ formal models of visual perception to create realistic graphics or compelling data visualizations; may try to identify patterns in society based on curating socioinformatic resources or at least make it easier for researchers

to create and use such resources. Digital Humanities broadens scholarship by making cultural resources more widely available. Moreover, web-based aggregation of humanistic research helps reveal patterns among the spectrum of humanities subjects and methods.

It is reasonable to say that the internal unity of human sciences can be more complex and loose than in many natural sciences — there is not necessarily a single set of a questions, a single calculation, or even a single clearly stated question which anchors the theory. Without obvious quantitative tactics, in many humanistic disciplines, researchers have a wider range of qualitative methods to explore, which also means that research communities become more fragmented, with the overlap of several blended theories situating each scholar in a small and very specific community. Arguably it is also easier, for the same reasons, for humanities researchers to find their own voice — co-authored papers seem to be much rarer in humanities than natural science, for example. But all this makes the metatheoretic dimension of humanities scholarship that much more important: each writer works within their own matrix of theoretical commitments, and we have to be sensitive to the threads which tie theories across different forms of argumentation, where a "single" theory, or single relatively consistent body of terms and concepts, is blended in different works with different other theories. The technological foundation for sharing, annotating, aggregating, and evaluating research is even more significant in the humanities than in natural science; and humanities scholarship is only scratching the surface of the potential for digital publishing and curation.

We are, however, witnessing an encouraging period of connection between humanities research and information-age sciences, spurred by a desire for more realistic and nuanced Natural Langauge Processing (NLP) engines; by recognition of a need for a spectrum of NLP technologies with varying degrees of difficulty (considering how real-world language can often be opaque to NLP ambitions) for varying text-analytic tasks; by a desire to inventory and catalog cultural resources and, in particular, to semantically manage nontextual content; by elegant applications of traditional philosophical subjects like Ontology, Mereology, and Modal Logic to data processing; and by the ideal of the Semantic Web, where alot of the information online is annotated with details to clarify their semantic meaning. As a result of these interdisciplinary projects, important branches of humanities scholarship are being formalized and codified in the hope of creating specific algorithms which manage text and data in flexible, practically useful ways.

Digital tools can serve humanities research in many ways — from data mining and analytics to document management and cultural heritage. Providing quality digital images for cultural artifacts, and a user-friendly, globally networked forum for sharing them, demands sophisticated software, web, and API development platforms. Technology is also increasingly relevant for advancing research into cultural objects — from 3D modeling to scientific tools for deciphering, imaging, dating, and performing chemical analysis to reveal locations and provenance; new technologies provide new capabilities to obtain information about objects. Science makes ancient texts readable, tells us where materials to create tools and works of art originated from, retracing trade routes and cutural contacts, or makes historical cities or habitations more alive by digitally reconstructing them. Perhaps soon Virtual Reality will give people a tangible feel of what it was like to walk into Rome or Athens, or live in a paleolithic settlement. Bioinformatics is yielding insights on ancient human migrations, and statistical comparative linguistics overlays this data with details of how languages spread and evolved. Considering the controversy in current times surrounding refugees, national boundaries, and ethnic homogeneity, it is important to remember how migrations and cultural contact were a defining feature of civilizations throughout history and before. Almost every human group migrated from somewhere else. We were all refugees once.

Along with these practically oriented projects, we should also not overlook the similarities between humanities and natural science from the perspective of Philosophy of Science. There are important differences but also important analogies between natural science and humanities *as* sciences. Both perspectival and methodological variation among human sciences, for example, derive from different scales of resolution for different research interests. A philosophical portrait of a single persons' experience, values, and identity addresses a different scale than economic and sociological analysis of group behavior. Variations in granularity mean than social sciences can be quantitative and data-driven for some research projects and Hermeneutic and case-study driven for others, with the finer and broader perspectives complementing each other. This inter-theory complementarity, which preserves theoretical autonomy but situates each theory in a larger space of knowledge, has many parallel in how multiple natural sciences overlap. There are also analogies in how methodology and subject matter complement each other: the techniques which yield warranted results in botany are different from quantum physics, for example, and this includes the role of instinct and intuition for the scientists who need to assess what to investigate, or how to make sense of what they see before them. One could find a parallel contrast between an economist and sociologist, a philosopher and anthropologist, a linguist and literary theorist.

Very tangibly, then, "hard" sciences like chemistry, genetics, and computational geometry are serving human sciences like anthropology and linguistics — not to mention how digital tools offer new avenues for artistic and musical creativity. But there is a more ideological dimension to the significance of natural science for human science: emphasizing the technical capabilities of modern societies, for example, is a way of embracing diversity over homogeneity, inclusiveness over exclusiveness. Prosperity is possible not by societies rejecting outsiders, but by trusting science to deliver health, nutrition, and a modern standard of living which can be widely shared. Refugees and immigration can be a benefit rather than a hidrance to modern societies because civil engineering provides the resources to absorb outsiders, allowing the benefits of diversity to come to the fore. Retrenchment and prejudice are therefore, also, forms of rejecting science. And moreover scientific truth, amongst truly committed to scientific principles, is not something consistent with prejudice or group-think: cultural traditions should be respected, but science cultivates an ethos of rational progress which is only consistent with communities' open-mindedness and willingness to evolve. Accepting science as a paradigm of reason and deliberation therefore carries with it a preference for open-minded, flexible, tolerant communities that adapt their norms and lifestyles to others' perspectives and to new scientific discoveries; and it can cultivate a politicized pressure on communities in general to embrace a similar open-mindedness. Science in reality, as an institution, does not always evolve along these benevolent lines, but such a spirit or rational open-mindedness is a logical consequence of the scientific world-view, which is more reason for science to be embraced and defended, including challenging science when it violates its own best instincts.

At the same time, science has much to learn from humanities. Most scientific theories explain phenomena which are far removed from everyday experience — in space, time, and scale. We arrived at that point gradually: to understand the medium-sized earth-bound objects of our day-to-day lives, we have needed to push to ever remote regions, galaxies and organic molecules and atomic nucleii, and the conditions of the early universe. But the conceptual matrix which preconditions scientific investigation and then makes sense of the results comes from normal experience — physics explains light, electricty, and gravity, but those things have meaning for us because they are inescapable features of pre-scientific conciousness. Life, desease, happiness, and language are all concepts emanating from a universal human experience that only gradually became subsumed by scientific theory. Science needs an inner awareness of how, without this grounding in everyday life, science reduces to mathematics, which is important in its own right but does not have a comparable anchor in an empirical world.

This is more than just stressing scientific "popularization", the need for scientific literature to include accessible summaries as well as rigorous articles — that is important, true, but the dependence of science on everyday life is deeper than that. The basic and most important concepts of science would literally not exist if it were not for a human mind already attuned to inertia and force, movement and stasis, singleness and complexity, heat and cold, light and dark, transparency and opacity, sickness and health; concepts manifest in literature and thought as well as science, from ancient times on.

Not all scientific concepts have this commonsense rootedness; in the inventory of scientific reason it is important to identify which concepts are formally derived from the basic human condition, which conversely are more technical, and how the former are transformed by the definitional rigor and mathematical formalization applied to them. In some cases science studies formally at a fine scale what becomes everyday experience at coarser scales: we do not see photons and light waves but are well acquainted with light itself, that emerges as the higher scale from electromagnetic phenomena. Living things, also, are emergent properties of complex biological systems. In other cases everyday experience glimpses larger-scale phenomena at a distance: it takes great ingenuity to build modern astronomy from the specks of light we see as stars. We perceive the past state of Earth from scant traces in fossils and geologic strata. Still, the empirical building blocks of many sciences are ones which people can tangibly grasp: the stars in the sky, fossils in the ground (and on display in museums), experiments that people can see live. The "raw data" of modern science can be much more ethereal, like bubble chamber diagrams which prove the existence of theorized particles, like the Higgs boson. But historical threads connect these more ephemeral empirical givens with the tangible objects and experiments which would have been considered empirical foundations for science in earlier ages, and these connections to real-life evidence should not be neglected.

It would also be wrong to understand science only in empiricist terms: while raw data is an essential part of formulating and testing theories, an equally significant aspect of scientific reason is building theories which compensate for missing and controversial data; for designing apparatuses which obtain data and discern which data is relevant; for building a coherent model out of empirical observations than makes predictions about data not yet obtained. Science is characterized by a search for data to confirm an exciting programme, like String Theory, as much as by a search for theories or mathematical models that explain observations already made. There are, in other words, two different axes characterizing scientific empiricsm: the relation of *a posteriori* and data-driven theorizing on the one hand to *a priori* and mathematics-driven theorizing on the other; and within empirical observations the relation between familiar and tangible givens, like fossils and the visible night sky, with indirectly obtained givens, like spectra derived from chemical analyses of fossils and radio-telescopic pictures of the sky. In abstract, mathematical models, also, there is a division between purely structural results and a groundwork which needs human intuition and common sense to fill out. Once a certain set of equations, or the definition of a number system or algebra, is formally defined, many solutions and properties follow out of structural necessity; it may take clever analysis to prove the results but our ascent to their correctness should not be based on intuition or mental pictures. Yet intuition has a distinct place in relating mathematic systems to things in the world: a greek symbol may stand in for an object or a force field but it takes a cognitive leap to grasp the mathematical structure, with all its provable properties, and then "attach" the abstract system to tangible things in the world — to see how the forces and limitations governing behavior of posits in the abstract system are in fact telling us something about the behavior of the associated concrete phenomena.

Science, in short, weaves between common sense and radically unfamiliar worlds; between abstract structural reason and mental pictures or intuitions; between tangible real-world data and phenomena which are understood as "observations" only by select experts. A Philosophy of Science must trace

how and where science does rest on everyday experience and intuition as well as where it supplants common sense; both are part of science. But consequently Philosophy of Science needs a well-reasoned understanding of "everyday experience" and "common sense" itself; these themselves are loaded concepts, and have social, cultural, and even political dimensions. Whose experience is "everyday"; whose sense is "common"? These questions cannot be left unexamined in a complete treatment of scientific reasoning.

This suggests that science is not complete without incorporating questions and forms of investigation nurtured within the humanities — while, at the same time, humanities are not complete without considering science. There is, at one level, an often unstated assumption that subjects of main concern for humanities — belief and value systems, political and economic decisions, the sense of personal and collective identity and its expression in conscious experience and practial action — have causal/physical dependency on phenomena of science, such as brain states and the biology of embodiment. But also, apart form that, understanding and refinining humanities methodologies benefits from how these issues have been worked through, often in more formalizable (and for that reason rather simplified) ways in natural science: emergence and the merger of theories addressing different phenomenal scales; the notion of empirical evidence in its different forms for different scales and types of analysis; the role of insight and intuition as compared with the role of data and quantitative analysis; the nature of theory-languages and how theoretic terms are understood to refer to things in the world. The questions of *whose common sense*, *whose subjective judments* or *whose criteria of objectivity*, *whose data*, *whose intuitions*, are no less pressing in humanities and social science than in "hard" sciences.

Like the spectrum of empirical data in natural science, from the tangible to the indirect, there is a spectrum of empiricity in human sciences where "observations" are sometimes hard data (social, economic, linguistic, and so forth) and sometimes more subjective and interpretive (case-studies, first-person accounts, writer's own introspective or restrospective judgments). All of these forms of data have a role to play, but how observations serve theories — and how theories explain data — has to be rigorously articulated within different branches of humanistic analysis no less than in natural science, and the approaches which Philosophy of Science takes in addressing these issues can serve as an example for the humanistic side. In short, explanatory and philosophical completeness brings the natural sciences into the province of the human sciences, and vice-versa.

Once this interdependence is fully recognized, it should lead to a reorganization of how knowledge and education is curated, and the idea that human and natural science are somehow opposed or in contrast needs to institutionally fade away, replaced by a greater sensitivity to both similarities and differences among and between sciences and humanities overall. Hopefully a byproduct of this evolution will be a greater fusion in the "intellectual culture" of these academic fields, which can lead scientists to a greater appreciation of the political importance of science to global democracy and human progress, and lead humanists to appreciate the Democratizing potential of theoretical rigor.

Appreciating the specificity of theoretical niche and the analytic overlap of diverse theories — covering a spectrum of granularity and methodology that can reinforce rather than undermine an overall understanding — this path to theoretical plurality, via a focused emphasis on theory-building, can help humanities become more inclusive, allowing each scholar, writer, and researcher to speak with their own voice. These are academic and metatheoretic ideas that can hopefully resonate outside academia, and ensure that humanities and natural science collectively can promote democracy and justice: a politicization of science which is also a scientization of politics.

Finally, perhaps tools discussed or aggregated here can modestly serve this goal, helping to gradually piece together a more scientific framework of humanistic scholarship via APIs and Semantically

Annotated writing, for example, and simultaneously a framework for deploying science concretely in the service of the public good, via Community-oriented data sharing and entrepreneurship platforms, Green technology, STEM educational resources, and facilitating political organization to defend science and promote science as an intrinsic feature of democratic societies.

8 Contact

Thank you for visiting this site, which is still under development.

Apologies therefore for its rough edges, and hopefully there is material here which is interesting or even useful despite them.

Comments, questions, suggestions, and additional resources to link or include here, would be greatly appreciated!

For now, please send comments to axiatropic.semantics@gmail.com