Web Science and Reflective Practice

Kieron O'Hara and Wendy Hall

Intelligence, Agents, Multimedia Group
School of Electronics and Computer Science
University of Southampton
Highfield
Southampton SO17 1BJ
United Kingdom
{kmo,wh}@ecs.soton.ac.uk

Abstract: The notion of reflective practice has been influential in professional practice of all kinds, including engineering. Reflective practice suggests that knowledge and practice are linked, that improvisation based on actual practice, as opposed to the application of formalised theory, drives understanding in many applied fields. A key feature of reflective practice is a feedback loop between actions and their effects, which need to be evaluated to produce understanding. The WWW is of course a piece of socially-embedded technology, and in many ways its progress can be seen in terms of such feedback loops, as development of formalisms and applications have social effects which are evaluated to create new opportunities for innovation. However, as we shall describe in this chapter, the WWW is embedded very deeply, and innovation can create social effects on extreme scales – for example, the creation of the practice of blogging and the massive linked blogosphere, which appeared in a very short space of time. The discipline which is called *Web Science* is intended to address the problem of reflective practice in a space where the feedback loops which facilitate learning happen at large scales over small time periods, increasing the danger that learning will happen at the cost of large-scale social damage (or damage to the WWW itself). In this chapter we will interpret the practice of Web engineering through the lens of the reflective practice concept, using the development of the Google search engine as a case study, in order to articulate the particular issues that individuate Web engineering from other types of engineering.

Introduction

The World Wide Web has become an extraordinarily transformative technology. Claims for its significance range from hype to scepticism, but in general most agree that its capacity for supporting communication and access to documents is orders of magnitude beyond previous technologies, and it has wrought great changes not only over the Internet and ICT, but also in the offline world, affecting the media, entertainment, politics and e-government, science and research, administration and commerce. Not only that but whole new areas of activity, such as social networking and multiplayer games in persistent virtual worlds, as well as new types of crime, have flourished using its protocols. The number of users is vast and growing, yet the decentralised structure of the Web – there is no editor of content, no quality control and anyone can link to anything – has democratised communications in all sorts of ways.

Yet for all its powerful effects on societies for good and ill, the Web is remarkably under-studied and under-theorised. There seem to be three principal reasons for this. First, it is a dauntingly large and complex structure. Second, it changes very quickly, so data soon become outdated. Third, it is a curious amalgam of technologies (hardware, protocols such as HTML and HTTP, and programming environments such as JAVA and AJAX) and human activities (the Web links not only documents and data, but people as well) and so a comprehensive overview demands multi-disciplinary skills ranging from computing, law, economics, sociology, management and organisation studies, media studies, semiotics, mathematics, as well as innumerable sub-disciplines. Too often the Web is studied as an example of a particular phenomenon – a network, or a set of computer languages, or a platform for commerce – whereas taken as a whole it is so much more.

Furthermore, the Web is not an exogenous entity. As Karl Marx once said, "philosophers have only interpreted the world in various ways: the point is to change it." Surprisingly, many have studied the Web without thinking that – if they did not like what they discovered – they could influence its development in more positive directions. It is an engineered technology, and so can be altered for the better. Conversely, many engineers have tried to change, and succeeded in changing, the Web, but if those changes are uninformed by an understanding of what the wider consequences are there is an element of risk. The internal risk of breaking the Web, somehow preventing or disincentivising the links which make up the Web network, should also not be discounted.

To this end, academics are coming together to foster a new discipline of *Web Science* (Berners-Lee et al 2006, Shadbolt & Berners-Lee 2008) to develop methods and curricula to understand the Web and provide foundations for engineering methodologies so that the Web can be changed for the better. If Web Science delivers a greater understanding of the Web, current threats can be identified and addressed, current opportunities pursued, and the Web itself can be adjusted to take account of social change.

In this chapter we consider some of the methodological difficulties of assembling this transdisciplinary amalgam of analysis and synthesis, study and engineering. Analogous problems arise in many engineering disciplines, so this is hardly untrodden ground (the 20th century planners who wished to create 'cities of the future' were grappling with problems of similar scale and complexity), but we will argue that the Web poses particular problems for its engineers as a result not only of its scale, but also of the range of scales at which it can be characterised. We begin in the next section by looking at methodological considerations, while in the following section we move on to consider how Web Science must operate, mapping Tim Berners-Lee's ideas onto a plausible engineering paradigm, and determining points of tension. We will put these abstract ideas in a concrete setting by looking at a large and complex Web phenomenon, Google.

Methodologies for studying the Web

The two major problems with the development of the Web Science paradigm are the creation of a common vocabulary and a common methodology across the relevant disciplines. Methodology is the focus of this chapter, and in this section we will discuss some of the relevant background to the investigation of the Web. In the first subsection we will discuss a commonly understood model for engineering practice which is not borne out by experience. Secondly, we will discuss some important points about computer science. In the third subsection we will look at issues surrounding transdisciplinary research such as that which will be required for the Web, while finally we will introduce the important ideas behind reflective practice.

Technical rationality

One common misconception about the relationship between engineering and science is that the latter is prior to the former. The model of technical rationality assumes that, given agreement on framing a problem and on the desired ends, the job of the engineer is to apply scientific theory to achieve them: the means are determined by a scientific engagement with the proposed and agreed ends.

This is a false picture in many ways, not least historically – the growth of technology and industry in the 18th century was independent of the development of science and mathematics, and the theories that explained the important processes of the industrial revolution did not appear until decades afterwards (O’Hara 2010, 127-130). Furthermore the application of science to technological and engineering problems is not as straightforward as the model suggests. The model of technical rationality relies on three distinctions which in practice are very hard to draw: between means and ends, so that a technical procedure (means) can be applied objectively to a pre-established goal; between research and practice, so that theories can develop in isolation from their application; and between knowledge how (procedural knowledge) and knowledge that (declarative knowledge), so that action can be derived from theory.

The main issues are twofold. The complexity of engineering-problem-solving means that (a) framing a problem and (b) framing a potential solution space are extremely difficult, socially embedded, affected by all sorts of practical, cultural, institutional and financial constraints, and obstructed by the constant political dialectic between competing interest groups which will arise in a free public space hosting a plurality of values. Framing the problem is plagued by extreme uncertainty, and the characterisation of the solution space characterised by conflict.

Complex engineering problems are more often than not in effect unique, not fitting easily under generalisations, meaning that the application of abstract theory is non-trivial. Meanwhile, no problem can be considered in isolation, and any potential solution will have unintended consequences elsewhere. Neatness is an unusual property of problems; as Russell Ackoff, one of the founders of operational research, argued. “Managers are not confronted with problems that are independent of each other, but with dynamic situations that consist of complex systems of changing problems that interact with each other. I call such situations *messes*” (Ackoff 1979, 99).

Computer science

With respect to the Web, one would naturally expect to consult the discipline of computer science which like Web Science amalgamates analysis and synthesis, investigation and engineering. After all, the Web is a piece of computing technology defined by protocols and formalisms such as URIs, HTTP and HTML. However, it should be borne in mind that where the Web is concerned what counts is their *use*, not their form. URIs are a naming convention, providing strings of characters to identify resources that are the targets of hyperlinks – but that they are *identifiers* depends on people identifying things with them. HTML, the Hypertext Markup Language, would not be of general interest unless real people in the offline world marked up documents with hyperlinks. The structures and expressive resources of these protocols are essential, but equally essential to the Web is the fact that they are used. The use of these systems cannot be described by the systems themselves, and does not fall within the purview of computer science.

Computer scientists do not have the disciplinary expertise to explain why the protocols have been used as they have, or what effects additional protocols would make. The tools of computer science, such as formal verification techniques for example, can only give a partial picture. Deriving behaviour from specifications is hard enough, but in the case of the Web the specifications of formalisms tend to be developed painstakingly by international committees under the aegis of the World Wide Web Consortium (W3C – http://www.w3.org/) A rough specification of a requirement (e.g. 'a language for expressing ontologies') will be hard to map onto a formalism expressive enough for the task (e.g. OWL Web Ontology Language).

In the case of OWL, its developers were canny enough to understand that users would vary in their demands, and so developed it as a series of three sub-languages, ranging from OWL Full, very expressive but non-computable and without constraints to preserve consistency, to OWL Lite, computable and constrained but relatively inexpressive (McGuinness & van Harmelen 2004). The existence of these various sublanguages shows awareness that the development of, in this case, an ontology depends not only on formal properties, but also on the informal purposes and institutional constraints on developers. Will the ontology be made up of flat hierarchies, be used a lot or demand regular editing and maintenance by non-experts? In that case, OWL Lite is the sensible option. If, on the other hand, the knowledge to be modelled is highly complex, well-understood and represents a hard-won global consensus, or if the modellers need the full expressiveness of the underlying knowledge representation language RDF (Resource Description Framework), then OWL Full is indicated. The point is that these actual operational details of the OWL language are social, economic and organisational – they are not computer science issues. The distinctions between the three sublanguages were created because of the perceived heterogeneity of ontology developers' demands, but beyond this recognition the computer scientist *qua* computer scientist has little to say. It may be, for instance, that OWL Full remains unused because too expressive and providing too little constraint – that would not be the fault of its developers, who could not be expected to anticipate in detail the demands on it.

To take another example which we will explore in more detail later in the chapter, Google's success derives largely from the brilliance of its PageRank link analysis algorithm (Page et al 1999), which determines the relative importance of each element in a linked network. This is a very impressive piece of work, and there is much about the algorithm for the computer scientist to get her teeth into. However, this only accounts in part for its success. First of all, the computer scientist can describe the recursive definition of 'importance' that emerges from PageRank (an important page is linked to by lots of important pages), but the congruence between that operational definition and the requirements of Google's users is an extra fact. To give just one extraordinary example, in 2007 a man who had been missing for some years turned up at a police station in the United Kingdom in an apparently amnesiac state. His disappearance had actually been an insurance scam, which was uncovered by a member of the public Googling him to find a picture of him taken after his disappearance with his wife (Weaver 2007). Nothing about PageRank tells you, however closely you look at the brilliant details of its weightings or damping factors, that it can be used for solving missing persons cases or insurance frauds in under a fifth of second.

Secondly, PageRank has to function not in the neat world of a formal definition, but in the real world with hostile elements who will try to subvert the algorithm to ensure their own unimportant content receives a high ranking. This Google-spoofing (as it is called) demands a whole layer of adjustments to the algorithm, in an arms race between Google and the spoofers. Third, it is not only the adjustments to the algorithm that count, but also the extent to which they can be kept secret, so organisational issues such as security also become important. Relevance, context, security – these are not from the computing vocabulary, and require a wider disciplinary focus.

Transdisciplinarity

Individual disciplines work via the notion of abstraction. In computing, for example (Colburn 2004, 322-325), important conceptual tools include data abstraction, procedural abstraction and language abstraction (i.e. the ascent from assembly language to higher-level languages). Abstraction entails universalisation and idealisation of diverse objects and relations but there is a serious question as to how far idealised theories and models can address or explain complex concrete problems – this is what theories of technical rationality kept bumping up against (Schön 1983). There is a mismatch between academic knowledge production and the knowledge needed for solving embedded social problems, between knowledge supply and knowledge demand.

The need to weave approaches together to address complex, dynamic real world environments (or to borrow a term from Husserl, the life-world, *Lebenswelt*, the world that subjects may experience together) has led to the idea of *transdisciplinarity*. Transdisciplinarity is characterised by (i) the need for a radical integration of the disciplinary work involved, including linking abstract and case-specific knowledge, (ii) the involvement of a large number of stakeholders in problem definition and solution specification, (iii) transgression of disciplinary paradigms, and (iv) a focus on problem-solving rather than theorising which (v) promotes what is perceived to be the common good (Hirsch Hadorn et al 2008). Methods need to be tailored for highly complex problems and the difficulties of integrating the various problem-solving approaches.

To this end, transdisciplinary research demands three layers of knowledge. *Systems knowledge* describes and models empirical processes, mapping onto the scientific knowledge assumed in the model of technical rationality, but it must be supplemented by *target knowledge* about the needs and interests of the practitioners and stakeholders in a problem, and *transformation knowledge* about the technical, social, legal, cultural and other means of acting to transform the existing situation and to create new technologies, structures or practices (Hoffman-Riem et al 2008, 4-5). Transdisciplinary research is a complex amalgam of activities, but some such perspective is required in order to bring diverse disciplines together to work in a complex *Lebenswelt* in whose progress many people have an interest.

Reflective practice

The criticisms of technical rationality and the requirements of transdisciplinarity each point towards the need for more knowledge, different in kind as well as quantity, about an engineering problem. Because of the *sui generis* nature of many engineering problems – this certainly applies to the Web of course – much of this knowledge, which is typically procedural rather than declarative, must be derived in practice, often in response to unforeseen challenges perceived during a project itself. This has led to the development of a theory of engineering practice called *reflective practice* (Schön 1983).

In this methodology, the problem as initially set is not fixed in stone, as the practitioner must change her perceptions and strategy in response to uncertainty, instability and unique features of a problem. She proceeds experimentally, but not, as in the scientific context, using the logic of confirmation. Rather the logic is of affirmation; the aim is not to raise hypotheses to falsify them (as with Karl Popper's falsification logic of scientific discovery) but to create and discover new solutions that need be neither unique nor optimal. Controlled, reversible experiments are out of the question, and so each experiment that the engineer tries must as far as possible be sensitive to the needs of the context, and take into account elements of the target knowledge and transformation knowledge – the systems knowledge cannot be tested in isolation.

The method of reflective practice is an answer to the requirements of Web Science, to improve the Web in its relation to the offline world. Indeed, given the complexity of the problem space, it will be essential to develop engineering methods that use the insights of reflective practice, dynamically and recursively reconfiguring the problem specification as more knowledge is gained during the design and engineering processes themselves. In the next section, we will discuss this cyclical view of Web Science in more detail.

Web Science: reflectively engineering the Web

Engineering the Web has been described by Berners-Lee as requiring sensitivity to both technical and social concerns, as shown in Figure 1. The designer has an idea for an innovation and develops protocols, formalisms, software and hardware to realise his vision or idea, which may or may not be formally or precisely specified. However, no digital system lives in a vacuum, and its use will depend on a number of assumptions that the designer makes about the social context. If those assumptions are incorrect, then the system, however technically adequate for its task, will not perform that task. Note also that the designer cannot specify every single aspect of the system's behaviour; at some point the assumptions about social context will have to carry some functional weight. The designer's assumptions about context are an essential part of the design. For instance, email systems have been developed in response to ideas about how the connectivity of the Internet could be used as a communications network. SMTP was developed on the basis of assumptions about what people would want the communications system to carry, about organisational context and about the motives of senders (specifically that messages would be sent in good faith by a homogeneous community all of whose members would be concerned with a group of problems determined by an organisational context, so messages would be relevant to the receiver, generated in response to a genuine requirement, with a transparent meaning).

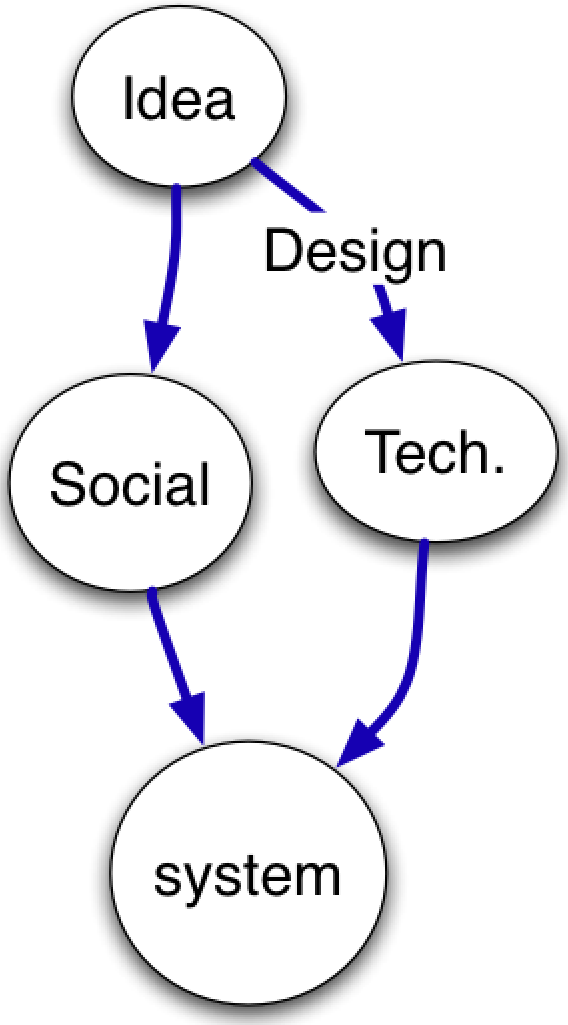

Figure 1: The Web design process (all figures in this chapter from Berners-Lee 2007)

This is a general view of engineering. However, the Web has particular issues associated with it, because the system does not live in a vacuum any more than the design. Figure 2 shows what happens as the system is let loose in the online environment. The Web itself has many hundreds of millions of users, and billions of pages and connections, so any system can result in emergent phenomena undreamt of by the original designers, whose social assumptions can hardly be expected to be accurate in the general case. Figure 2 shows the idea of Figure 1 being implemented with some technical work and a set of social assumptions, to produce a micro-level adjustment to the Web environment, but if enough users take up a system, there will be a marked and noticeable change in macro-level perceptions. It may be that older patterns of behaviour change, or that they are supplemented by new behaviours, or that new users swell the online community (for example, consider the growth of the blogosphere, and how this has changed not only the Web, but also the media, journalism, politics, commerce and social interaction). However that may be, the end result is in effect a new Web understood at the macro level, as a result of micro-level engineering.

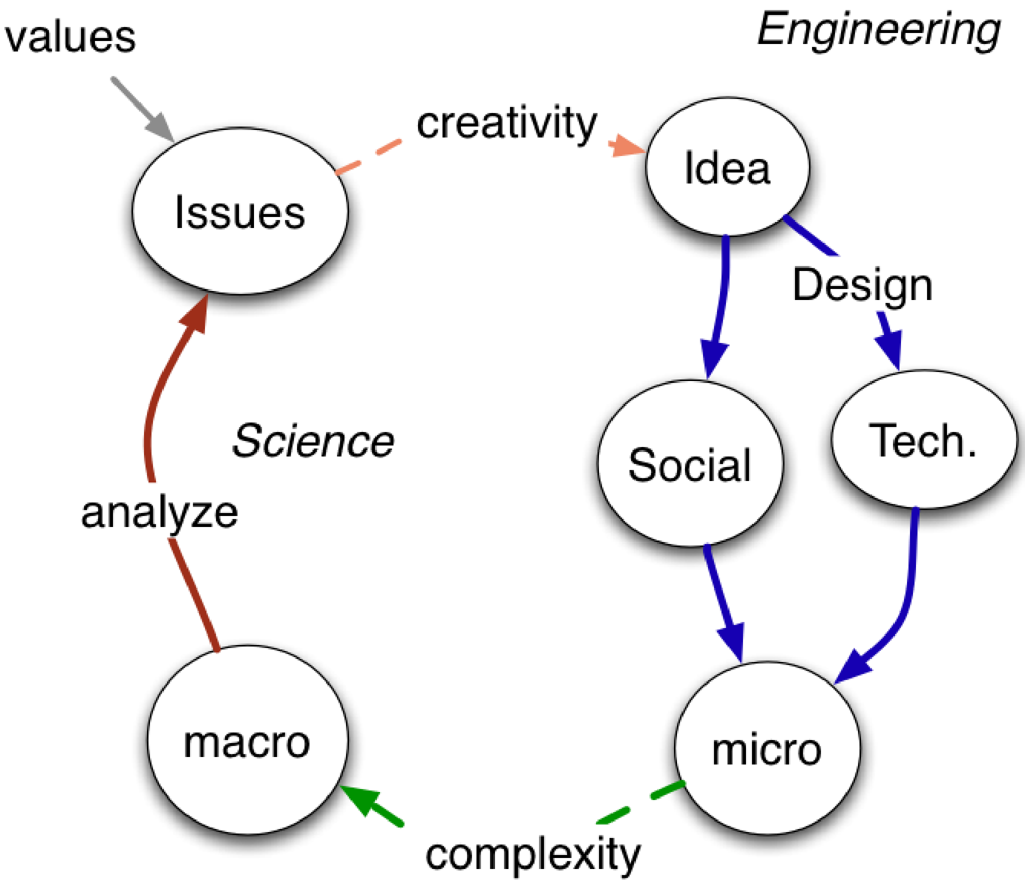

Figure 2: The Web and the world

Those macro-level effects will need to be analysed in order to understand their wider implications, and this process of analysis will throw up important issues which will resonate either positively or negatively with the values and needs of the embedding offline societies. Unresolved issues will pique the interest of entrepreneurial designers, whose creative efforts will lead to new ideas which then have to be designed and engineered on the basis of a new set of social assumptions and the cycle begins again.

To continue the example of SMTP, the invention of email was predicated on a set of social assumptions that at implementation time were realistic (the system was largely used within the scientific community whose main concern is usually with the sharing of knowledge), but when it became a macro phenomenon, used not only by members of the target community for unforeseen types of communication but by people not in the target community at all – and ultimately the number of non-academic users has dwarfed the number of academics – the unintended consequences of a free and simple communication method became clear. Problems such as spam and phishing began to emerge. These were seen at best as nuisances, at worst as torts or frauds which raised important issues. New technical solutions, such as spam filtering, were now needed to solve the problems created by the emergent phenomena associated with the new technology. Social changes also accompanied the technology. Emails leave a semi-permanent record so it became harder for companies to hide their internal decision-making (an important factor in the prosecution of Enron, whose preserved emails now constitute a fascinating data source for those interested in corporate communication), while divorce lawyers now regularly sequester years-worth of communications from errant spouses. These developments have demanded adjustments in the law, corporate best practice, and our intuitive understanding of privacy which themselves raised more issues, and so the cycle continues.

To be sure, the Web is not the only artefact whose unintended consequences can be vast; a badly-placed bridge in a city can affect all sorts of economic relationships. Nevertheless, the disparity in scale between the micro engineering and the emergent macro phenomena is especially characteristic of the Web, whose sheer size not only spreads problems across the globe, but also creates new problems by taking design solutions out of their intended social context.

So far, Berners-Lee's characterisation of Web Science (2007) as a cyclical conversation between scientists and engineers, users and techies, fits neatly into Schön's (1983) ideas about reflective practice, while arguments parallel to Schön's show that engineering the Web cannot be a matter of technical rationality. However, his framework can also be used to point up the singularities of the Web as a piece of designed technology which demand its intensive study as a first order object as envisaged by the Web Science programme.

One point which must be made immediately is that most professional disciplines receive much of their developmental impetus from real-world requirements that are not always immediately obvious to the eye. For example, medicine has as its nominal goal wellness, but as Foucault and others have argued the meaning of 'wellness' has evolved, sometimes quite rapidly, and has had a range of interpretations from being able to live a life with minimal personal aid, to the elimination of pain, to approaching an ideal of the healthy body which itself might be determined by the medical profession or by society. However, quite aside from the nominal goal, a great deal of medical expertise is actually concerned with the problem of litigation – how to avoid it, and how, if it strikes, to construct a rationale for and defence of one's actions.

Web Science currently lacks this behind-the-scenes motivation. Its nominal goal is to improve the Web. This cannot be left solely to market structures and the straightforward profit motive; the Web is an arena for amazing innovation, but not all the innovations have been benign (and certainly their effects are rarely accurately predicted even by the startup companies that promote them). Yet there is little agreement on what constitutes an improvement of the Web. A sterner constraint – imagine a class-action suit in the US courts brought by victims of spam – would dramatically focus the effort. Without such an unwelcome focus, Web Science must include not only a debate about how to connect technical and social developments empirically and conceptually, but also a parallel debate conducted with diverse stakeholders about normative requirements.

This is a minor problem compared to the way that Web Science problems are necessarily framed and addressed. If we consider the zone of time in which an action may make a difference, what Schön calls the *action-present* (1983, 62), which depends on the pace of activity and the boundaries of potential action, we find it is both tiny and vast, depending on point of view. The cycles of Web development are measured in years. Blogging, for instance, took a number of years to develop from small beginnings, and then 'suddenly' took off at the beginning of the century. 'Suddenly' in this case is still a matter of years from, say, the appearance of the first blogging tools and guides and the first major political issues influenced by bloggers in 2001 and 2002, to the exponential growth characteristic of the years after 2004. But what counts is the timescale of an effective intervention. The phenomenal growth of the blogosphere was predicted by very few (as Tom Wolfe quipped, one by one Marshall McLuhan’s wackiest predictions come true), and its specific effects on, say, political discourse or the offline media was anticipated by even fewer. The timescale is certainly large enough for technical development, but the social context evolves alongside the technical as well as driving it. What seems imperative in year 0 of a research project may be completely out of date by year 3 when a product appears.

So, for example, Twitter proved its political worth for many as an important conduit for news about spreading protests about the conduct of the 2009 Iranian Presidential Election, trumping traditional media outlets which were slow to feature the story. However, the downside of Twitter was also revealed at the same time when the story spread and the useful messages from inside Iran were lost in a tsunami of well-meant but pointless messages from America in support of the protesters. Furthermore, as well as describing, measuring and discussing the individual phenomena of an episode such as this, it must also be seen in the context of wider arguments, e.g. about the deleterious effect of an always-on media (Rosenberg & Feldman 2008).

New types of online behaviour become very popular very quickly. At the time of writing (2009), Facebook and Twitter dominate thinking about cutting-edge large-scale Web phenomena, but by, say, 2014 it is quite likely that the landscape will be very different and the giants of five years previously will be hopelessly out of date. Datasets for large-scale modelling are extremely important to alleviate this issue, and some projects, such as the EU project Tagora (http://www.tagora-project.eu/), have begun to explore these spaces retrospectively. Such analyses are clearly ways forward, but as each new star application comes along, new users (possibly responding to different incentives) will arrive with it, rendering old assumptions void. Not only that, but a five or six year development and growth cycle will take many of the most enthusiastic users from adolescence to adulthood with all the attitudinal changes that implies. In short, the scale of the phenomena means that what seems a relatively long action-present for Web Science is in reality very curtailed. By the time data are gathered, models created and simulations run, the opportunity to influence events may already be past.

As noted earlier, it is characteristic of large-scale engineering that controlled experiments are impractical because their effects cannot be restricted or reversed. The relation between change and understanding is different in reflective practice precisely because of this constraint. The requirement to understand is subservient to the requirement to change for the better (unlike in disinterested research, where understanding is an autonomous goal), but the ever-present danger is that an experiment makes a permanent change for the worse. An extra and unusual issue with respect to controlled experiments on the Web is the variance of scale between the experimental setup and the outcome. Any experimental change will be of relatively small scale – a new type of software, a new type of communications protocol. The consequences *relative to the intention of the innovation* can be described and studied in small-scale experiments in the lab, or with a small set of pioneer users. Such intentions are usually focused on the experience of a single user or a single organisation. The problem, of course, is that few if any of the massive global consequences of Web technologies are of this tractable type, because they affect very large groups of people and organisations, so that even the benign or positive consequences at the scale of the Web as a whole are unintended.

Engineering using reflective practice inevitably involves trade-offs between the consequences of an artefact in relation to its intention, and the full set of consequences both intended and unintended (Schön 1983, 153ff.), but the Web is an especially difficult case because the consequences in relation to the intention of the engineered development are relatively small-scale and detectable fairly quickly, while the unintended consequences, good or bad, emerge years later at a scale far beyond the control of a single person or corporation.

One way of expressing this mismatch is to look at three levels of analysis in the evaluation of design. First, the design specification includes a normative element against which it can be evaluated ("the artefact should do X"). Second, any design detail has to be evaluated against and be consistent with previous design decisions. Third, the designer must be sensitive to any new problems that arrive during deployment. As Figure 2 shows, the third level brings in phenomena at the macro scale which may take years to manifest themselves. So distributed and decentralised is the Web that even the second level is likely to be beyond the individual design team's capacity for understanding.

Another illuminating way of looking at it is through the lens of the transdisciplinarity framework as discussed above (Hirsch Hadorn et al 2008), which postulates three kinds of knowledge, of systems, targets and transformations. The Web engineer is possessed of the systems knowledge of the artefact being constructed (the node labelled 'tech' in Figure 1 and Figure 2). In those diagrams, the target knowledge (about stakeholders) corresponds most closely to the nodes marked 'social'. Hence the systems and target knowledge are, from the point of view of the individual engineering project, tractable. However, the transformation knowledge, of all the various systems relevant to implementation and use, is key, and yet is once more out of reach of the immediate designer. If we consider Figure 2, the transformation knowledge is relevant to the 'macro' phenomena on the left hand side, connected to the technical parts of Berners-Lee's diagram by a dotted line denoting the uncertainty of the connection between micro to macro phenomena.

To conclude this section, Web Science looks very like an example of reflective practice in engineering. However, the largeness of scale of the problem definition, the speed of the development cycle in comparison to the action-present and the massive disparity between the designer's understanding and the breadth of the relevant phenomena are extremely problematic. The relative absence of real-world datasets for investigation adds to the difficulties, but the retrospective nature of empirical research may mean that such investigations are anyway of only limited value relative to new systems. The need for a clear and deep integration of disciplines under the Web Science banner is evident, and is the focus of the Web Science Trust (http://webscience.org/), a research organisation dedicated to the development of methods and curricula for the analysis and engineering of the Web.

Case study: Google

How do these issues pan out in a real-world situation? Earlier in this chapter we discussed academic attempts to understand Google's PageRank algorithm as an example of the essential transdisciplinary complexity of Web-based issues. Let us now consider this company in a little more detail to see how complexity stands in the way of understanding, and how Google, in particular its founders Lawrence Page and Sergey Brin, have finessed the engineering problem (Battelle 2005, Stross 2008).

Google's general experience fits in rather neatly into Berners-Lee's Web Science lifecycle, as he himself has pointed out (Berners-Lee 2007, and see Figure 3).

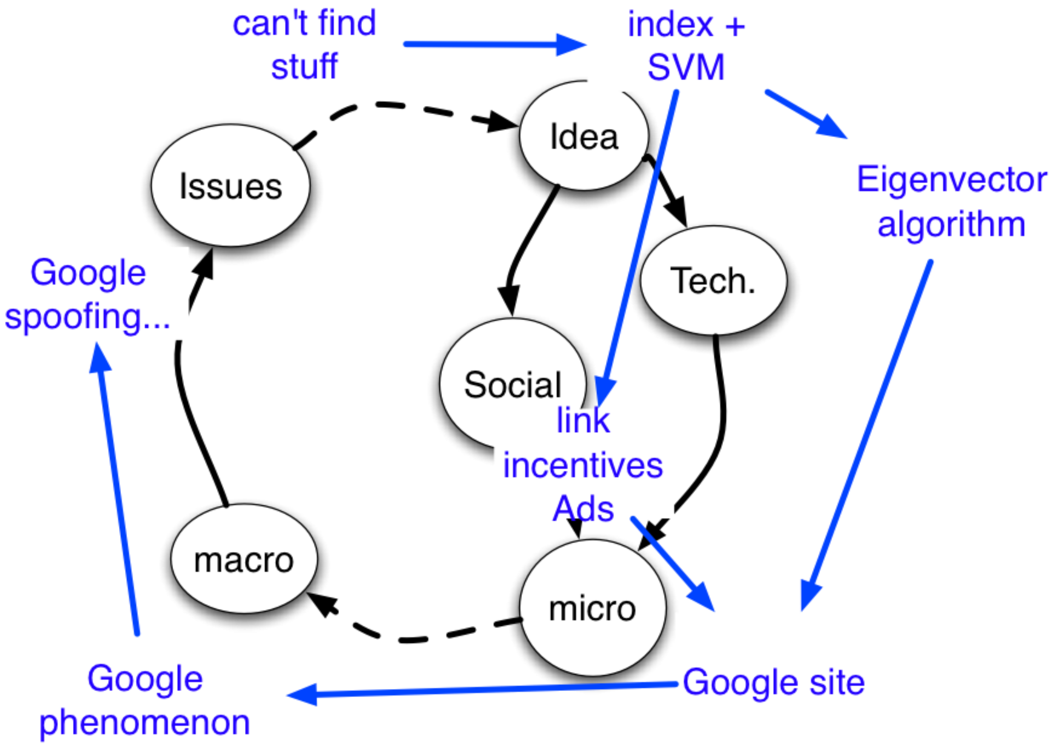

Figure 3: The growth of Google seen in Web Science terms

Here the initial issue that began the cycle was the growth of the Web and the difficulty the search engines of the mid-1990s had in scaling with it (their search methods were usually designed with a particular size of Web in mind, and typically would become inadequate as the Web grew beyond that scale). The initial idea was for an index together with a support vector machine, a type of multi-dimensional classification learning method. The PageRank algorithm, which uses a linear algebra method of computing the Eigenvectors of matrices representing the link structure of the Web, was the technical solution chosen, and this was expected to work in a social situation where Web authors were pleased to link to other sites they thought relevant. The result was the Google search engine, which of course grew to extraordinary size and influence.

However, new issues soon appeared. So powerful was Google that others found it began to pay to spoof PageRank, usually by creating lots of fake pages all linking to a particular page. As mentioned above, Google then had to take measures, both in terms of their algorithm and their corporate practices, to counter that. As we can see, Google's evolution fits neatly onto the schema of (Berners-Lee 2007). In the terms of transdisciplinarity, Google's problem has often been with the transformation knowledge – the requirements of the law and intellectual property in a variety of jurisdictions is often harder than the engineering (still less was Google prepared for the controversy surrounding its move into China). Yet the issues raised by Google go beyond Berners-Lee's brief diagrammatic analysis.

The company prides itself on its array of geeky talent, but it would be a mistake to assume that this is sufficient. In particular, Google relies upon the prevailing ideology of openness on the Web. If the Web becomes transformed by a preponderance of pay-per-view sites, subscriptions, walled gardens or proprietary software (Zittrain 2008), then Google's approach will be less powerful. The rise of Facebook in recent years, with its revivification of the walled garden model, has been perhaps one of Google's major challenges (Stross 2008, 21-46).

The ideological demand for openness has some surprising effects. For instance, it precludes Google's charging for search except in certain circumstances (in its early years it provided search as a commodity for Yahoo!). That means that it must differentiate itself on quality, which has enabled it to retain its market lead (Pollock 2008). Pollock presents an economic model of search in which users expect high quality search and refuse to pay, while search engines have the problem of finding a business model. Advertising is the key of course and advertisers want as many users as possible. In his model, this leads to a tendency towards a concentration of the market, or even monopoly (consistent with the history of the search industry so far). Monopoly does not necessarily mean that the public welfare is compromised – as long as search quality is retained, the antitrust issues may not be serious. Nevertheless accurate models are needed in order to provide predictive power, to anticipate the conditions under which concentration of the market might lead to a decline in welfare, and to inform technical and regulatory approaches to the Web.

Surprisingly, Page and Brin seem to have come late to the realisation that advertising was a potential basis for a business model (Stross 2008, 3ff). The beauty of search for an advertiser is that the user's search terms indicate exactly what he is interested in at exactly the point at which the advert is served up, and this certainly unanticipated consequence of the Google model has completely subverted the advertising and media industries, to the point that mainstream media are losing money and some of their more expensive functions (quality drama, quality journalism) are being undermined for lack of a viable business model of their own.

Commitment to quality has always been a key factor for Google. One of Page and Brin's insights was that quality demands the ability to scale up with the Web, and they have invested heavily in hardware. The result is immense growth in the data storage industry (Google has been influential more widely by its promotion of the paradigm of cloud computing, which centralises storage in giant data warehouses). Here again a development which of itself creates interesting problems in the abstract (what methods for search and retrieval will work over these giant repositories?) ramifies in all sorts of unexpected ways. Cloud computing creates enormous issues for privacy preservation, for example, or the legal jurisdiction under which one's data storage falls, but the issues go beyond software or even organisational and legal structures, to hardware. A large data centre consumes about as much energy as an aluminium smelter, and policies to address climate change will certainly have implications for the continued growth of the industry.

The point of this case study is to show how quickly an idea on the Web ramifies into other areas of importance. It is certainly not intended to be critical of Google (which is a defender of the Web and has an interest in preserving its ideology of openness), but merely to show how an idea produced an ideology, new types of cheating (not by Google – we refer to Google spoofing) an unexpected business model, a monopoly, disrupted two enormous industries and will have a tangible effect on the planet's climate in centuries to come. To predict the effects of Google's business decisions will require deep analysis integrated along transdisciplinary lines. Google is not the only example of course – see (Berners-Lee 2007) for more.

Conclusions

To conclude, the cyclic structure of Web Science is an example of the 'conversation with a problem' of the reflective practitioner. Technical knowledge (technical rationality, or systems knowledge) will not be sufficient to create a Web that serves humankind, though it *will* be sufficient to build a widget that makes the Web more dangerous, less open or less connected. The imperative for Web Science, therefore, is to try to bring together relevant expertise, to discover not only systems knowledge but also wider knowledge about stakeholders and the social context, to develop theories that evaluate designs on wider as well as more narrow criteria, and to involve as wide a range of people as possible in expressing and framing the problems it is intended to solve.

References

Russell Ackoff (1979). 'The future of operational research is past', *Journal of the Operational Research Society*, 30(2), 93-104.

John Battelle (2005). *The Search: How Google and its Rivals Rewrote the Rules of Business and Transformed Our Culture*, London: Nicholas Brealey Publishing.

Tim Berners-Lee (2007). *Looking Back, Looking Forward: The Process of Designing Things in a Very Large Space*, inaugural lecture, University of Southampton, http://www.w3.org/2007/Talks/0314-soton-tbl/#(1).

Tim Berners-Lee, Wendy Hall, James A. Hendler, Kieron O'Hara, Nigel Shadbolt & Daniel J. Weitzner (2006). 'A framework for Web Science', *Foundations and Trends in Web Science*, 1(1), 1-130.

Timothy Colburn (2004). 'Methodology of computer science', in Luciano Floridi (ed.), *The Blackwell Guide to the Philosophy of Computing and Information*, Malden MA: Blackwell Publishing, 318-326.

Gertrude Hirsch Hadorn, Susette Biber-Klemm, Walter Grossenbacher-Mansuy, Holger Hoffman-Riem, Dominique Joye, Christian Pohl, Urs Wiesmann & Elisabeth Zemp (2008). 'The emergence of transdisciplinarity as a form of research', in Gertrude Hirsch Hadorn, Holger Hoffman-Riem, Susette Biber-Klemm, Walter Grossenbacher-Mansuy, Dominique Joye, Christian Pohl, Urs Wiesmann & Elisabeth Zemp (eds.), *Handbook of Transdisciplinary Research*, Berlin: Springer, 2008, 19-39.

Holger Hoffman-Riem, Susette Biber-Klemm, Walter Grossenbacher-Mansuy, Gertrude Hirsch Hadorn, Dominique Joye, Christian Pohl, Urs Wiesmann & Elisabeth Zemp (2008). 'Idea of the handbook', in Gertrude Hirsch Hadorn, Holger Hoffman-Riem, Susette Biber-Klemm, Walter Grossenbacher-Mansuy, Dominique Joye, Christian Pohl, Urs Wiesmann & Elisabeth Zemp (eds.), *Handbook of Transdisciplinary Research*, Berlin: Springer, 2008, 3-17.

Deborah L. McGuinness & Frank van Harmelen (2004). *OWL Web Ontology Language Overview*, http://www.w3.org/TR/owl-features/.

Kieron O'Hara (2010). *The Enlightenment: A Beginner's Guide*, Oxford: Oneworld.

Lawrence Page, Sergey Brin, Rajeev Motwani & Terry Winograd (1999). *The PageRank Citation Ranking: Bringing Order to the Web*, Stanford InfoLab Technical Report, http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf.

Rufus Pollock (2008). *Is Google the Next Microsoft? Competition, Welfare and Regulation in Internet Search*, http://www.rufuspollock.org/economics/papers/search_engines.pdf.

Howard Rosenberg & Charles S. Feldman (2008). *No Time To Think: The Menace of Media Speed and the 24-Hour News Cycle*, New York: Continuum.

Donald A. Schön (1983). *The Reflective Practitioner: How Professionals Think In Action*, London: Maurice Temple Smith.

Nigel Shadbolt & Tim Berners-Lee (2008). 'Web Science emerges', *Scientific American*, October 2008, 60-65.

Randall Stross (2008). *Planet Google: How One Company Is Transforming Our Lives*, New York: Free Press.

Matthew Weaver (2007). 'Woman found canoeist photo via Google', *The Guardian*, 6th December 2007.

Jonathan Zittrain (2008). *The Future of the Internet: And How to Stop It*, London: Allen Lane.