BILETA

17th BILETA Annual Conference

April 5th - 6th, 2002.
Free University, Amsterdam.

Regulating Architecture and Architectures of Regulation: Contributions from Information Systems

Ian Hosein, Prodromos Tsiavos and Edgar Whitley
(Department of Information Systems, London School of Economics and Political Science)

Introduction

In a world where technology and politics interact, there is increasing discussion of the role of law and government action. This is particularly true in the case of cyberspace, where the internet is changing the ways in which regulation is applied. In the extreme case, technology is claimed to determine freedom and rights. This position has been addressed by Laurence Lessig, in his book "Code and other laws of cyberspace" (Lessig 1999).

In this book, Lessig discusses how regulation could occur, and does occur, within a new regulatory habitat (Hood 1994) where a technology's architecture (or "Code") acts alongside the traditional regulators of the market, laws, and norms. According to Lessig, Code can be an effective regulatory mechanism, but also represents the interests of its developers and is therefore socially shaped (MacKenzie and Wajcman 1999).

Although Lessig highlights the fact that Code is designed and hence he holds a position that runs counter to notions of strong technological determinism, he still seems to reify the notion of Code. Information Systems studies the development, implementation and use of computer based systems and offers an alternative reading of Code from that provided by Lessig. By exploring Code in more detail, it is possible to add the same level of sophistication to our understanding as Lessig has done about the other elements of his framework for regulation.

Examples of areas that Information Systems has studied that apply to Lessig's problem area include the differences between bespoke and packaged software (Carmel and Sawyer 1998) and the unintended consequences of large scale software implementations (Ciborra 2000). Other research has shown the implications of low level Code design decisions on organisations and industries. For example, Scott (2000) describes the transformation of risk in financial markets arising from the introduction of a new system to aid in loan decisions.

In this paper, we seek to develop questions of regulation through architecture by presenting two cases where the design of software systems affects the ways in which regulation can take place. First, we consider the design of peer-to-peer (P2P) networks for file sharing systems; second we review Microsoft's Cryptographic Application Program Interface (CAPI). In so doing we challenge the reified notion of architecture as a straightforward aspect of regulatory intervention.

In order to structure our critique of Code, we introduce three main ways in which to consider technology and regulation. These are the ways in which technology can object, the modalities of

regulation and the path of technology regulation.

Technology as a Regulatory Actor: Objecting

In both of the cases of regulation involving technology that will be presented in this paper, the construction and the constitution of the technology are important components in the regime. Within the literature on regulation however, the role of technology is often poorly attended to. While it may be acknowledged as a disruptive force, as Peltzman notes within telecommunications rate structures (Peltzman 1989 p.117) and interest rate regulation (1989 p.121), it is not investigated in detail. The need to consider technology in regulation is not new, however, nor is it particular to the internet and digital communications technologies; as Levin (Levin 1966) noted regarding the management of the radio spectrum in the 1960s: "New communications has required the reexamination of many policies and assumptions in recent years".

The ability to examine technology policies and regulatory change is limited, as studies usually concentrate on human action. Hood (1994) notes the many ways of seeing regulatory change: the power of interest groups, power of ideas, social transformations, or a policy's self-destruction (Hood 1994 p.4); these explanations are all assuming that regulatory change is enacted only by humans. As Peltzman notes on pressures for deregulation being (1989, p.108)

... changes in the `politics' and changes in the `economics' of the regulated industries. Political change includes such things as shifts in the relative political power of contending groups and changes in the underlying organization and information technologies

again technology appears to be secondary to powers of humans and group interests. Meanwhile, Porter and van der Claas (1995) acknowledge that a failure in some environmental regulations is the static view of technology, and they believe that ideal regulation would set the conditions for the creation of technology to aid environmental interests. All of these concepts fail to acknowledge that technology is a strong regulatory factor in itself, and they fail to look in detail at how technology may disrupt or promote regulation.

The disruptive capacity of technology has been noted previously within the social studies of technology literature. Technologies may disrupt human action through objecting (Latour 2000). Latour's view is that technologies can force us to renegotiate our paths and goals (Latour 1999), and so restructure our regulations. They can also resist our attempts to regulate and deregulate, possibly more so than humans and institutions.

Technology and Human Action: Habitats and Modalities

Technology as a regulatory factor is not a new idea; but requires interrogation. Often when technology is brought to the foreground for analysis of a societal issue, it is considered a deterministic force, ignoring the power of human action. In academic studies it is poorly investigated and interrogated, often left in the background of analysis. At best technology is considered as a quiet part of the policy habitat (Hood 1994), in what is considered a society-centred approach, where policy changes arise due "to the background changes in technology and social structure" (p.10). As the social structure or the technology changes, it is said, there is a resulting loss of policy habitat, resulting in regulatory change. Our goal is to find a way of looking at the policy habitat as a socio-technical phenomenon, where the technology can be brought to the forefront, with the social actors.

Such technological and policy habitat explanations of policy shifts are considered problematic for a number of reasons. First, we must avoid views where technology is seen as deterministic; regulatory change is not automatic from technological change (Hood 1994 p.12). Hood continues that even when similar policies are adopted by different societies, "it does not necessarily mean that they are responses to the same `functional' problems"; and "social change is not necessarily an independent

factor from which everything else stems--it may itself by the product of other policies, designed to `shape' preferences" (p.12). Hood concludes that previous attempts to explain the loss of habitat give too much credit to technology, "leaving too little room for the autonomous dynamics of politics". Again, society and technology theorists are well aware of these concerns; as this is consistent with the anti-essentialist theory that argues we must never assume that technology is an object in itself with its own capacities; but rather these capacities are always interpretations (Grint and Woolgar 1997). Technology and regulation therefore interact within the realm of politics; different societies will interpret the problems and the technologies differently. Change may (or may not) come about due to technological or political issues.

Therefore a richer view is required; technology is not deterministic, as it requires human interpretations and action. Yet, human action is not deterministic either as technology objects and also regulates human action through the loss of a policy habitat. Lessig tried to develop a view where technology, law, norms, and the market were all modalities of regulation (Lessig 1999 p.88) that constrain individuals.

There were times these other constraints were treated as fixed--when the constraints of norms were said to be immovable by governmental action, or the market was thought to be essentially unregulable, or the cost of changing real-space code was so high as to make the thought of using it for regulation absurd. But we see now that these constraints are plastic. That they are, as law is, changeable, and subject to regulation. (Lessig 1999 p.91)

Lessig establishes that all of these *modalities of regulation* can all be shaped; but in his zeal to dismiss determinism and to say that technology can be shaped (it is only plastic like the other modalities), he misses on two points. First, he fails to acknowledge that the other modalities may interpret the technology in a number of different ways. Which laws apply, how many definitions of a given application exist in law? For example, cryptography is in law as a confidentiality tool and a tool for digital signatures. Each modality does not necessarily understand technology in one specific way. Second, in his claim that the constraints are merely plastic, he does not acknowledge how technology may object, how it refuses to be shaped by the interests of the other modalities. This latter point will be discussed in greater detail in the next section.

Walking the path of Regulatory Discontinuity

When Lessig analyses *the four modalities of regulations* (Lessig 1999 p. 235) he makes a distinction between architecture and the other three modalities, namely law, the market and norms:

The constraints of architecture are self-executing in a way that the constraints of law, norms, and the markets are not (Lessig 1999 p. 236)

Analysing in more detail his idea of *self-execution* of architecture as a regulatory modality, Lessig provides two further criteria, namely *agency* and *temporality*, based on two perspectives:

that of someone observing when a constraint is imposed (the objective perspective), and that of the person who experiences the constraint (the subjective perspective) (Lessig 1999 p. 237)

From the objective perspective, Law and norms always have an *ex post* sanctioning effect: if you violate the rule, the sanction comes *after* your action. On the contrary, Market and Architecture have an *ex ante* preventive effect: you cannot perform the action at all.

From the subjective perspective, the adherence to the rules contained in each one of the four modalities the acting subject is internally prevented from doing a particular action. The more the internalisation process is advanced the less probable is that the person is going to perform the prohibited action and thus the more effective the regulatory modality is. For Lessig, Law and Norms

are dependent on the internalisation of the rule, whereas this is not the case for architecture. Quoting Sorkin, Lessig describes the essence of Code's regulatory capacity:

Whatever the sources of the content of these codes, (...) their consequences are built (Lessig 1999 p. 239)

Code has indeed a great regulatory capacity; at the same instance it is a highly idiosyncratic form of regulation and we cannot really assess its consequences unless we attempt to analyse it.

First and foremost, technology as a regulator is not deterministic. Although Lessig overemphasises the self-execution of the code as a regulatory modality he also acknowledges that there is no certainty that a technology will produce a particular behaviour; rather it will affect it. (Lessig 1999 p. 239) This realisation has a profound effect, as it highlights the subjective or soft element of architectural regulation. Lessig claims that "architecture can constrain without any subjectivity" (Lessig 1999 p. 238), but in the same book he urges us to realise the "plastic" nature of technological constraints (Lessig 1999 p. 91):

There are choices we could make, but we pretend that there is nothing we can do. We *choose* to pretend; we shut our eyes. We build this nature, then are constrained by the nature we have built (Lessig 1999 p. 234)

The regulatory nature of architecture starts long before it is in place. The regulatory nature of architecture lies beyond its "artefactual" manifestation and is deeply routed in human subjectivity as can be seen in the quotations of different users of peer-to-peer technologies that we present in the case study.

The construction of the regulation does not soley have to do with the code; it is inherently linked to the use and implementation of the code as well. As the Actor Network Theory theorists have often emphasised, "reality is a process" (Callon 1986) or to use Latour's (1996) aphorism "For technology, every day is a working day". We need to move beyond the mere observation that technological constraints are "plastic" and constructed and try to explore the mechanics of this construction.

To advance the argument on the non-deterministic nature of technology we need to understand that technology has always a series of unintended consequences that sometimes have a greater impact than the original intended plans of its creators. (Ciborra 2000, Latour 2000).

Case Study 1: Napster

In May 1999 Shawn Fanning, an undergraduate student at Northeastern University, created an application called Napster. The idea behind Fanning's software was to enable end-users to share the MP3 files stored in their computers, using a centralised indexing service to locate the files. Two years after its launch, Napster had experienced an exponential growth to reach an audience of over fifty million users. Napster's popularity resulted in a lengthy legal battle between the music industry and Fanning's newly founded company on issues of copyright infringement.

Perhaps unsurprisingly the Napster case contributed towards the software's notoriety and led to the establishment of "Napster" as a generic term for "peer-to-peer" networks. Fanning's file-sharing service, however, was neither the first peer-to-peer (P2P) technology used for file-sharing, nor was it a pure peer-to-peer application. Indeed, what is meant by peer-to-peer is unclear:

Taken literally, servers talking to one another are peer-to-peer. The game Doom is peer-to-peer. There are even people applying the label to e-mail and telephones. Meanwhile, Napster, which jump-started the conversation, is not peer-to-peer in the strictest sense, because it uses a centralized server to store pointers and resolve addresses. (Shirky 2000)

If Napster was not a "pure" P2P application and it is not clear what P2P really is, then why insist on using the term for explicating the regulatory properties of technology?

Regardless of the technical nature of Napster, it has been inextricably linked to the term peer-to-peer and this has led to the labelling of a whole family of technologies as anti-regulatory, anti-authority devices. The social and conceptual construction of peer-to-peer technologies as anti-control mechanisms was a collective process facilitated in part by the media industry and the high profile legal action against peer-to-peer services that made them an icon for teenager users and thus triggered their proliferation, sophistication and anti-authority status.

Most of the peer-to-peer services have been entangled in legal disputes with the media industries. Napster was sued in 1999 and a shortly thereafter Scour faced a very similar fate. Both companies had to suspend services (from July 2001 and December 2000 respectively). In a mist of a series of legal developments and extensive media hype, the Recording Industry Association of America (RIAA) and Motion Picture Association of America (MPAA) have also gone after a number of other peer-to-peer services based on the most advanced peer-to-peer technology available at the time: that provided by FastTrack. Morhpeus, KaZaa, Xolox and Grokster have respectively been the targets of the media industry both in the US and in Netherlands. Xolox was shut down in the process, but Morpheus and KaZaa have survived to become the two largest P2P networks. RIAA and MPAA have also expressed their intention to prosecute individual P2P users.

Commentary on Napster

The whole process of the interaction between the authorities and P2P technologies can be seen as an example of the "cockroach phenomenon" (Schema I): After the filing of lawsuits and the subsequent shutting down of Napster and Scour, a new generation of P2P services came to the fore. Many of these had already been around for some time; however, it was the enforced suspension of Napster that made users migrate to them. The so-called "Napster Diaspora" soon became as large as the original "Napster Community". Although metrics for P2P use are not readily available, it is estimated that about 50 million users have downloaded Morpheus and over 35 million have the KaZaa desktop.

The RIAA and the MPAA responded with a second round of lawsuits against companies using FastTrack technology. The result was not entirely successful. Some of these companies were shut down but those that survived gathered most of the remaining users. In addition other services that were not under legal prosecution appeared and currently host over 75 million users. In a sense the legal battles functioned as a catalyst in the process of the physical selection of different technologies and disturbed the existing P2P "ecosystem"; the closure of Napster caused users to seek alternatives; by publicising the whole process (and once you start a legal battle in the media sector it is impossible not to go public) internet users were made aware of P2P technologies and the user base increased.

Most importantly, however, the phenomenon is both quantitative and qualitative. The new P2P systems were not simply more; *they were different from their ancestors.* Only the more technically sophisticated and legally-resilient systems could survive the RIAA / MPAA attacks. Moreover, all the P2P systems that came into existence during or after the Napster and FastTrack cases, irrespective of whether they had been a direct target of the media industry or not, encapsulated the characteristics of the systems that survived those legal actions: they had a bigger degree of de-centralisation of the indexing service and they were either open-source or located in P2P-friendly jurisdictions.

Another unanticipated consequence of the legal action against P2P systems was the emergence of the P2P *concept* itself. P2P existed as a technology for many years before the appearance of Napster. However, as we have already indicated it was the music-file sharing systems that popularised the technology and it was the hype surrounding the legal actions against it that gave P2P its current

notional content and an anti-authority focus.

However, by emphasising the anti-centralisation and anti-hierarchical attributes of P2P technologies, it is possible to adopt a rather one-sided view of the whole issue. Whilst P2P technologies have characteristics that defy sources of external control, at the same time they are very much about "awareness of the self and the others" (Gong 2002); they are about sharing, interconnectivity, parallel processing, communication and collaboration. Each of these goals can only be achieved through the use of rules governing the operations and relations between the nodes of P2P networks. That is, by their very nature, P2P networks are self organising and self regulating:

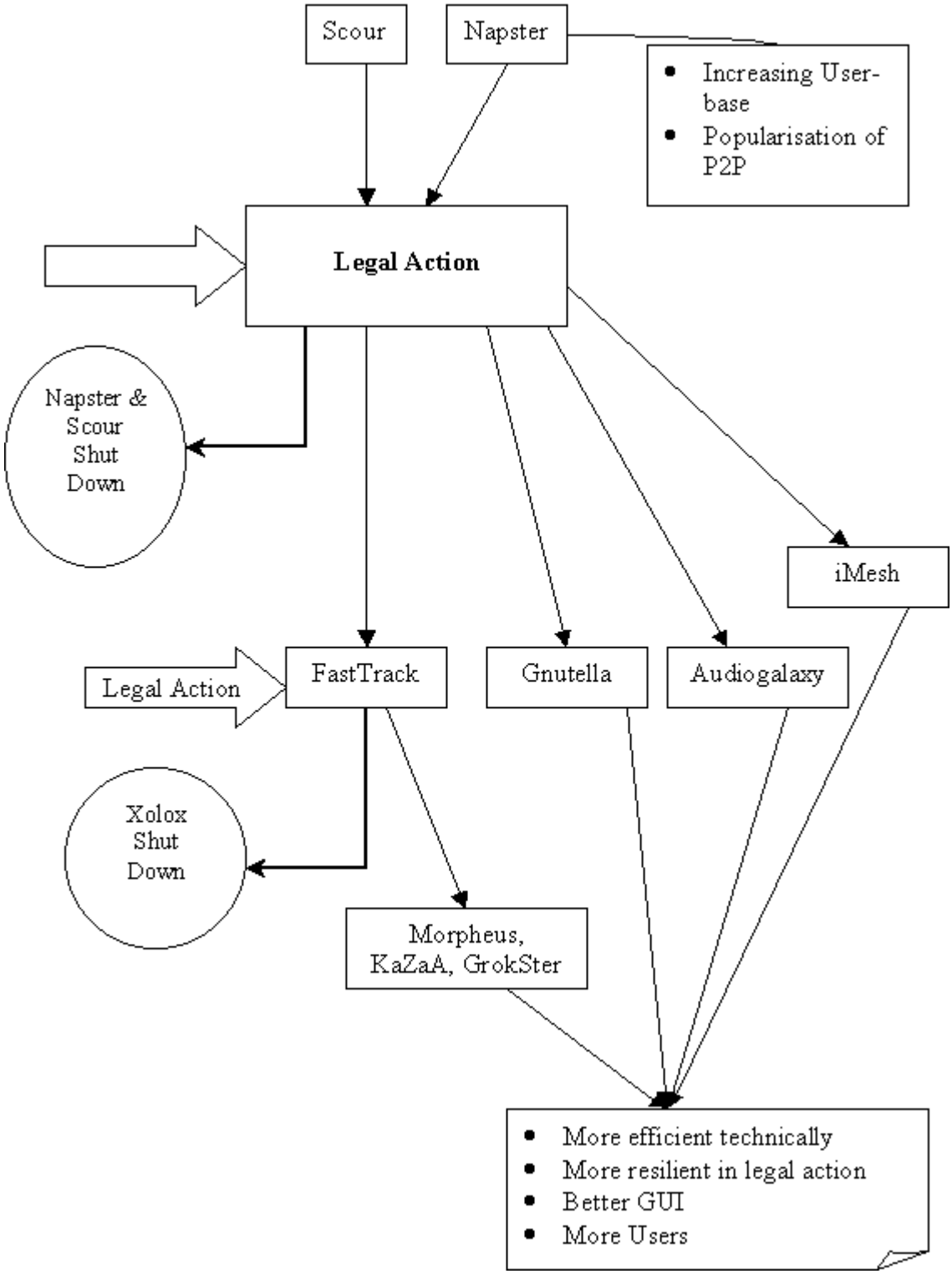
Obviously much remains to be done before P2P establishes itself as a lasting force. System monitoring, remote peer control, usage metering, and accounting methods are just few of the areas that need further research (Gong 2002)

Thus the continued successful development of P2P networks will require the accommodation of more self-regulatory mechanisms. The technology that is the architect of freedom to share files becomes a technology that enforces self-regulation.

In a very elementary level these mechanisms are needed for the solution of seemingly pure technical problems. It is nevertheless impossible to distinguish the regulation of technical issues with the regulation of human behaviour, especially when the latter occurs within the framework of these systems.

For instance, almost all the second generation file-sharing systems allow the users to regulate the amount of downloads and uploads that will take place in their part of the network. Also, in order to retain their operation many of the P2P file-sharing service providers install monitoring software.

The whole issue requires further research to identify and explicate the regulatory characteristics of P2P systems. At this point we just need to emphasise the fact that efforts to externally control technologies as disruptive as P2P systems are inherently flawed and can only lead to the cockroach phenomenon. At the same time, the position that P2P systems are anti-regulatory or anti-control is too simplistic and crude to be accepted. They encapsulate regulatory characteristics that control both technology and human behaviour. The point is not whether regulation exists or not but which is its locus and content.



(Schema I) The Cockroach Phenomenon

Case Study 2: Microsoft, export controls and CAPI

Throughout the 1990s, the US has revised its controls on the development and proliferation of

cryptography, the means of establishing confidential communications and data to ensure integrity and authenticity (Schneier 1996). The primary mechanism for controlling cryptography was export control regulation on products. This section describes the regulatory environment in the 1990s in which Microsoft developed its cryptographic solution, CAPI, up to 2000.

The substance of the regulations gradually changed from crude export controls on "munitions" to a more sophisticated understanding of the nature of the technology. In particular, the relative *strength* of cryptography implemented in products has been controlled by requiring a reduced *key size*--the relatively unique numbers that lock and unlock the data--to a size that can be *brute-forced*--guessed within a reasonable amount of time. While recommended key sizes for symmetric key cryptography are at least 128-*bits* long, i.e. 2^{128}, the US Government regulated the size of exportable key-sizes to between 40 and 56-bits, in various cycles of deregulation. These shorter key-sizes are far easier to brute-force than the 128-bit recommended standard.

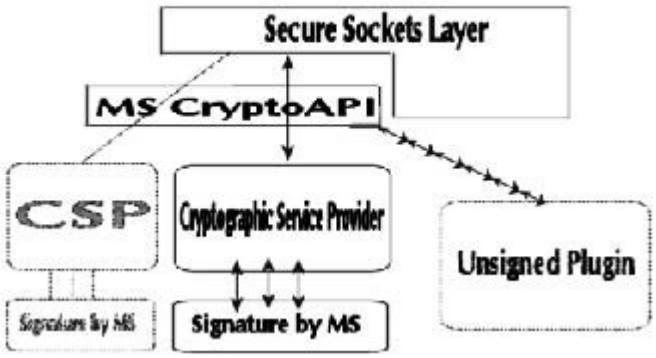
One consequence of the cycles of shifting regulations in the late 1990s was to create an unstable environment for code writing, especially for commercial packages. A company, such as Microsoft, which hoped to sell its products in many different markets needed to produce many different versions of the software to comply with the export controls that applied at any given time.

For many software developers, this required developing two pieces of software: one for domestic use where 128-bit+ keys were enabled, and another for export where reduced key sizes were enforced. Distributors with international markets were in an inconvenient situation, and this was sometimes costly in terms of production and public relations (e.g. (Laurin 1997)). An alternative was to sell one version with *weak* encryption for both domestic and international use.

Microsoft was susceptible to these concerns, having to cater for both the US export controls and consumer concerns regarding on-line security[1]. To deal with the regulatory environment, Microsoft embarked on finding a solution that could be designed and implemented in all its products, but that also satisfied export restrictions.

Dealing with such problems is common in software development, and the standard solution is to separate out the functionality that changes from that that remains relatively stable. By providing a standard interface to the variable aspects of the system, developers are isolated from the changing environment. This can be seen, for example, in the design of internet software, where the design of a web page is done without any consideration of the means of transmitting data over various cables and networks.

Microsoft implemented its solution to the problems of dealing with changing regulations through the development of its Cryptographic Application Program Interface (CAPI) that acts as standard interface to cryptographic software. By using CAPI, the developers of an application simply had to call the appropriate routines without worrying about how they were implemented. The cryptographic modules, or Cryptographic Service Providers (CSPs) achieve the flexibility and could be adapted to address the changing export controls by including different key sizes, algorithms and operations to be processed.



(Schema II) CAPI with two signed CSP plug-ins, and one non-signed CSP.

The benefit of this design was that one set of code (CAPI) is implemented for worldwide distribution, and then export restrictions would apply to the CSPs. Microsoft removed the liability for export by implementing cryptography into CAPI, as the functionality is enabled by CSPs, that are not necessarily Microsoft products.

Commentary on CAPI

However, not just any CSP would work with CAPI. The architecture of the system was refined to allow for a particularly interesting form of regulatory intervention. In particular, the US Government was concerned that outside software developers would not be able to write their own CSPs that would work with CAPI. It therefore required that CAPI first check that the CSP it was planning to operate with was an approved piece of software.

(e)xport control laws not only constrain cryptographic and related products but also any products which are specifically designed to interface to, or integrate with, cryptographic products. In effect, therefore, the very principle of openly available (CAPIs) is in direct conflict with the existing export control provisions in many countries. Thus, to integrate a CAPI into their operating systems without making them subject to export control Microsoft has had to establish some rigorous CAPI control procedures (Gladman 1996)

CAPI was thus designed so that it would only work with CSPs that are digitally signed and verified by a public signature key embedded within CAPI. In order to be digitally signed in this way, however, the CSP had to be approved by the US government. An unsigned CSP or forged-signature CSP would not operate with CAPI because the CSP authorisation-verification procedure would fail.

As Gladman makes the case for Microsoft,

It is important to recognise that this situation is not of Microsoft's making. In publishing and promoting a CAPI for use with their products Microsoft has gone as far as it can under US law to establish an improved basis for the provision of cryptographic information security when using their products. The procedures (...) are the provisions which the United States administration has imposed in order that Microsoft can offer their operating systems in world markets without being subject to US export controls (Gladman 1996)

Every CSP developer would have to gain approval for export from the US government, and only then would Microsoft digitally sign the CSP.

Without such a signature requirement, there is no way for Microsoft to guarantee a CSP is staying within export guidelines. Because unrestricted access to CAPI would make Windows ineligible for

export, the signature requirement limits CAPI access to vendors that agree to implement in conformity to US law (Kerstetter 1998)

The signing of the CSPs may be interpreted differently, however. According to Microsoft,

The primary purpose of the digital signature is for the protection of the system and its users: by signing the CSP the integrity of the CSP can be guaranteed to the operating system. The operating system validates this signature periodically to ensure that the CSP has not been tampered with. (Microsoft Corporation 1996)

The signing process, therefore, appears to act both as a gatekeeper for Microsoft and as a regulatory-enforcement mechanism for the US government.

The regulatory burden is thus placed on the CSP developers. First they need to get a CSP software developers kit (CSP-SDK); if a developer is outside of the US, they must apply for a license, which is in turn also regulated by the US Government[2]. The foreign developer would still have to acquire US government approval before the CSP can be used with CAPI. Gladman summarises this situation

CSPs Produced in North America for Domestic Use

The CSP Software Development Kit (SDK) is freely available without export control.

Microsoft will sign a CSP module without US (or other) government involvement

CSPs Produced in North America for Export

The SDK is freely available without export control.

Microsoft will sign a CSP module given evidence of United States government export approval.

CSPs Produced Outside North America

The SDK is subject to US government export control.

The Microsoft signature on a CSP is deemed to be a `defense service' provided by Microsoft to an overseas supplier and as such it is subject to the provisions of United States export control laws.

Adapted from Gladman (1996)

In effect, US export controls applied to all products that are designed for CAPI, whether they were developed in the US for export, or developed in Europe. This extra-jurisdictional reach is a design implication of CAPI, and invites the US Government into the functional loop of authorising the operation of applications with Microsoft Operating Systems.

While there are no indications that CAPI was designed to lock out non-North American CSP developers, that they needed to get approval by US export regulators was an obvious hindrance. If foreign CSP developers wished to make strong cryptography available to non-North Americans, the recommendation articulated from Microsoft, as documented and analysed in (Gladman 1996), was,

"For suppliers who want to maintain the same product across all markets, North American and everywhere else, the most attractive strategy remains to develop CSPs outside the US or Canada and outside (CAPI)". This is again a clear recognition on Microsoft's part that it will NOT be possible to use their CAPI to support the general availability of strong cryptography outside the United States

and Canada

Gladman concludes that Microsoft expected the US government to use CAPI to limit the development and use of cryptographic capabilities outside the US.

In terms of practical effect the mechanisms for the control of CSP signatures will be used by the United States administration to extend the scope of US export controls to cover CSP modules produced for domestic use in other countries even when there is no legal basis for such domestic control either in the United States or in the country concerned (Gladman 1996)

Under that export regime, while North Americans had open access to strong cryptography with Microsoft products, "the rest of the world will have nothing of any real value except in specialised application approved by the United States administration" (Gladman 1996) Gladman later noted (1999) that a UK office was later set up for the signing of CSPs.

Conclusions

With technologies increasingly developing regulatory features the boundaries between regulation and enforcement are gradually blurring. Are we regulating technology or are regulated by technologies? Taking up Lessig's call for research, "In cyberspace we must understand how code regulates--how the software and hardware that make cyberspace that it is regulate cyberspace as it is" we have presented two cases that will help us overcome the reified notion of Code.

These are now analysed using the themes presented earlier for refining our understanding of code and other forms of technology.

Technology as objectors:

Peer-to-peer systems may be seen as technologies that object to or resist regulation, even as companies fold to legal pressures. CAPI may resist deregulation; that is, even after the US export controls changed under pressure from lobbyists and privacy advocates and thus Microsoft would no longer be required to review the CSP export permits, CAPI would continue to require the digital signature of Microsoft to function, still disallowing all now legal non-signed CSPs. As a result, technology can act to bring about change through objection and resist deregulatory efforts even as human institutions bend.

Habitat and modalities

Lessig wished to counter the technologically deterministic view that the internet cannot be regulated. Instead, he leaves us believing that technology is merely plastic, and by extension, can be shaped by the other modalities of regulation. While his conception of the policy habitat is rich, he does not allow for technology's ability to object to new policies, nor its ability to sustain intentions that are not embedded by the market, laws, or norms. Nor does he allow for how the other modalities of regulation will interpret the technology in different ways (e.g. the market views cryptography differently than the law). CAPI's requirement for a Microsoft signature across jurisdictions regulates regardless of the market, the laws, and the norms.

The path of regulation

Moreover, there is no such a thing as a unique monolithic technology. Instead we are always faced with families of technologies that develop their internal dynamics and evolve in unanticipated ways. For instance, Audiogalaxy may impose a kind of regulation by installing by default a spy-ware application on the user's hard drive. However, technologies like Ad-ware, allow the user to override the spy-ware applications. If the norm of Audiogalaxy and Gator is "you have to pay with your

personal data for the services we provide", the norm of Ad-ware is "information is free, personal data has to be protected".

The "cockroach phenomenon", which we described in the case study, illustrates that there is no certain outcome in the interaction between different modalities of regulation, neither is one modality totally separate, independent or more effective than the others.

Revisiting Lessig's conceptualisations about technology as a regulatory factor, we see the mirage of control to be collapsing. Indeed technology is a modality of regulation, but it is one that cannot really be separated from the other modalities neither does it make any sense without them; it is not clear who is the creator of this regulation as it contains inscriptions from various (and sometimes undefined) actors; the content of the regulation is not always clear and it keeps changing through time; it does not have a deterministic effect and sometimes it is even autonomous.

Is technology indeed a form of regulation? We hold that it has *regulatory characteristics*, but the question of whether it constitutes a modality of regulation remains open. Tracing the regulatory characteristics of technology is exploring a path of discontinuities. Even if technology *is* regulation, it is of a very special kind. Before we have traced and analysed some of its characteristics we cannot say anything truly meaningful about it.

Following from Lessig and Hood, we believe that in the foreground of our techno-regulatory analysis we need to have both societal and technological factors, interacting, interpreting, and objecting. If we see regulation as a socio-technical issue, with human actors inscribing interests into regulation and technology, interpreting the technology, but the technology being able to object, then we may better understand how a policy habitat is developed, sustained, and lost.

References

Callon M (1986) Some elements of a sociology of translation: domestication of the scallops and the fishermen of St Brieuc Bay. In *Power, action and belief: A new sociology of knowledge?* (Law J ed.) 196-233, Routledge & Kegan Paul, London.

Carmel E and Sawyer S (1998) Packaged software development teams: what makes them different? *Information Technology & People* **11(1)**, 7-19.

Ciborra C U and Associates (2000) *From control to drift: The dynamics of corporate information infrastructures.*(Oxford University Press, Oxford).

Gladman B (1996) *US Government Controls on the Microsoft Cryptographic Application Programming Interface: A Paper for the ICE Workshop* International Cryptography Experiment, The Third Workshop 7th Draft

Gladman B (1999) *Re: NSA key in Windows* Personal communication September 6, 1999

Gong L (2002) Guest editor's introduction: Peer-to-peer networks in Action. *IEEE internet Computing* 37-39.

Grint K and Woolgar S (1997) *The machine at work : technology, work, and organization.* (Polity Press, Cambridge, Mass).

Hood C (1994) *Explaining Economic Policy Reversals.* (Open University Press, Buckingham, England).

Kerstetter J (1998) Crypto holes slow export adoption. *PC Week*

Latour B (1996) *Aramis, or the love of technology.* (trans. Porter C) (Harvard University Press, Cambridge, MA).

Latour B (1999) *Pandora's hope: Essays on the reality of science studies.*(Harvard University Press, Cambridge, MA).

Latour B (2000) When things strike back: A possible contribution of science studies to the social sciences. *British Journal of Sociology* **51(1)**, 107-124.

Laurin F (1997) Secret Swedish E-Mail Can Be Read by the USA. In *Svenska Dagbladet,* 18 November 1997 Archived at http://catless.ncl.ac.uk/Risks/19.52.html

Lessig L (1999) *Code : and other laws of cyberspace.*(Basic Books, New York, N.Y).

Levin H J (1966) New Technology and the Old Regulation in Radio Spectrum Management. *The American Economic Review* **56(1/2)**, 339-349.

MacKenzie D and Wajcman J (1999) *The social shaping of technology* (Second edition) *(*Open University Press, Buckingham)

Microsoft Corporation (1996) *Government Regulation of Cryptography*

Peltzman S (1989) The Economic Theory of Regulation after a Decade of Deregulation. In *reprinted in A Reader on Regulation, 1998* (Baldwin R, Scott C and Hood C eds.) (Oxford University Press, Oxford).

Porter M E and van der Claas L (1995) Toward a New Conception of the Environment-Competitiveness Relationship. *The Journal of Economic Perspectives* **9(4)**, 97-118.

Schneier B (1996) *Applied Cryptography.*(2nd edition) Wiley and Sons,

Scott S V (2000) IT-enabled Credit Risk Modernisation: A Revolution Under The Cloak Of Normality. *Accounting, Management and Information Technologies* **10(3)**,

Shirky C (2000) *What P2P is and what isn't* O'Reilly P2P Archived at http://www.openP2P.com/pub/a/P2P/2000/11/24/shirky1-whatisP2P.html

---

[1] Some countries also had import controls such as France where they existed until 1999

[2] As an aside, exports are regulated by the Department of Commerce, but SDKs are regulated by the State Department as a munition.