Distributing Justice in a Digital Society

Nagarjuna G.*
Homi Bhabha Centre for Science Education, TIFR

October 2018

This article is on the politics of media, covering both digital and non-digital media. The primary purpose of this short article is to bring home the point that the current political powers, all around the world, are using the 'new' digital medium to make 'old' governing systems more powerful, rather than empowering citizens, although most of the governments of nations themselves claim to sustain and work towards democracy. We will discuss centralised regulation and decentralised regulation of media, and relate this to the concepts of copyright and copy*left*. A political movement has taken shape to address the issue, which is called *free software movement*, and this has inescapable implications to several aspects of our lives, wherever digitisation of culture impacts and effects.

Historically, as the digital form of information and communication technology (ICT) unfolded, two cultures developed, those who used the new medium and those who abused the new medium, and these have evolved into two incommensurable, indeed, sharply polarised communities. In the light of the emergence of the modern information society, while the policy makers, philosophers, social scientists have been caught napping, power-hungry agencies, an umbrella term that includes governments and mega-corporates, have taken advantage of this lapse.

This essay attempts to identify the ontological aspects of these new forms of technology, economy, and politics, to explicate both the might and plight of the new digital natives. In the process, the roots of the game and the anatomy of the game played by the 'mafia' as well as the masses are clarified. The discourse is situated in the context of some case studies, related to digitised data, knowledge, software, spectrum and ICT infrastructure.

Introduction

There exist at least two kinds of managing systems. Let us call the first as central-control-model (CCM) and the other as distributed-control-model (DCM) (or decentralised control model), for want of better names.

The CCM is well established, and is commonly considered to be an acceptable form of control in civilised societies, and the latter, DCM, is taking shape

in the new digitally and technically mediated social space. Considering that most polities today are democratic, one may wonder, why do we assert that the CCM is the most common? The fact of the matter is DCM is the most talked about political design (or desire), but remained largely on paper, until recently, as we begin to see the design being implemented successfully. Perhaps sufficient conditions for democracy are only just getting satisfied, after the emergence of the 'new media' without the 'mafia.' We will draw the implications of the DCM for policymakers.

CCM is the most common means of managing a social system. However, the reality is that maintaining CCM is expensive and vulnerable, while DCM is economical and sustainable. One may well question why, if CCM is expensive and vulnerable, how did it come to be recognisably the most common? It is important to note that most of the political or administrative systems, including the modern democracies, remained a CCM despite the professed objectives of democracy. But why?

The answer is, we suppose, that we learned how to take power from the commons, but did not learn how, in return, to grant power to the commons. As a result, those in power have come to believe that the ignorant and illiterate commons would misuse power. Therefore the 'ignorant', 'uneducated' commons needs to be protected from the wise, educated elite. Although, in theory, democracy developed as a bottom-up model of polity, in practice it remains top-down. The so-called 'democracies' seen in practice seek power through votes from the commons, but power itself is kept under wraps, and centralised. People elected a president or a prime minister, but effectively only through a hierarchy of electoral colleges, and this assures the centralisation of administration. Ironically, the wrapped up power itself requires security and protection, exhibiting the vulnerable state of the centralised batteries of power.

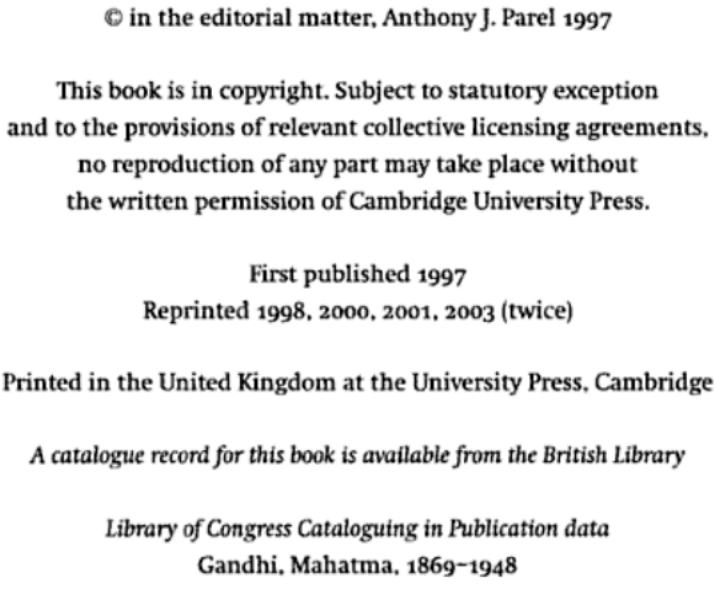
In what follows we will illustrate these contrasting models concerning media politics which, we hope, will offer an insight to formulate our policies.

Copyright vs Copyleft

Soon after the industrial revolution, copyright was established as a social instrument, to protect the power of authors or creators. (Pollard, 1922) (Cohen & Rosenzweig, 2006). Although copyright was institutionalised as an incentive to authors, it soon transformed itself as an instrument to protect the agents of authors, the publishing houses.

As is required by copyright law, a statement that describes the conditions of copying the creative work is stated following the date and name of the holder of the copyright. While such statements change from publication to publication, in almost all cases the copyright holders assert a variety of ways in which readers are restricted, in the use of the published material.

To understand how this works in the modern form, let us look at the copyright page of Mahatma Gandhi's *Hind Swaraj*, republished by Cambridge University Press. Even the most ardent followers or scholars of Gandhi do not



© in the editorial matter, Anthony J. Parel 1997

This book is in copyright. Subject to statutory exception
and to the provisions of relevant collective licensing agreements,
no reproduction of any part may take place without
the written permission of Cambridge University Press.

First published 1997
Reprinted 1998, 2000, 2001, 2003 (twice)

Printed in the United Kingdom at the University Press, Cambridge

A catalogue record for this book is available from the British Library

Library of Congress Cataloguing in Publication data
Gandhi, Mahatma, 1869–1948

Figure 1: Copyright page of Hind Swaraj republished by Cambrdge University Press. Mahatma Gandhi, the author of the book, asserted "No rights reserved" for the Hind Swaraj.(Achal, 2012)

know that in fact the book was published initially by Gandhi under the assertion "No Rights Reserved." It should be recalled that he was a lawyer, and was certainly aware of the ramifications of making such an assertion. In practice, the re-publishers of the same book do not respect the original author's claims, which is clear in Figure 1. By inserting editorial inscriptions, a preface or an introduction, the publishers republish the work in a different *avatar* and use this thin excuse to claim copyright of the work. Thus even in cases where the author did not create restrictions, publishers have found ways to prevent the expression from reaching readers without restrictions. This illustrates how blatantly the proprietary media houses have exploited a legal instrument.

The usual restrictions on copying can be illustrated from another example, a book published by MIT Press, titled "Philosophy of Computing Information" has this on the copyright page: "© 2004 by Blackwell Publishing Ltd. (citation) All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by the UK Copyright, Designs, and Patents Act 1988, without prior permission of the publisher." Such statements are a norm rather than an exception.

As the history of its usage suggests, almost everyone in the world has used copyright to restrict the way the resource can be, or, in effect, cannot be used by others. At the time that Gandhi sought to make a difference, the seed did not

find fertile soil. However, when Richard M. Stallman (popularly known as RMS) invented copyleft, by turning copyright on itself, the ground was fertile this time. Several other authors, mostly software programmers, employed copyright the way RMS did. The following example illustrates this contrasting use.

The book that contains selected essays of RMS has the following: "© 2002 Free Software Foundation. Permission is granted to make and distribute verbatim copies of this book *provided the copyright notice and this permission notice are preserved on all copies.*" (Italics added.) As we notice, this is a creative subversion of the legal copyright instrument. It permitted, not restricted, copying. However, it imposed a condition, that all copies must carry the same copyright notice. This simple act of granting freedom to copy, provided the freedom is not withheld in future by others, is called copyleft (R. M. Stallman, 2002). Though the above copyright statement shows 2002 as the year, RMS invented it around 1983. (R. Stallman, 1985)

Copyleft is a paradigm case of how DCM works. It distributes responsibility, by granting the freedom to copy to everyone. It provides an incentive for the care and responsibility the receivers promise to take. The returns are inbuilt. The recipients get more than what they return. Since they do not have to impose any restriction, no additional energy is required to be spent. Most publishers, on the other hand, control their property by employing lawyers specialising in intellectual property to tailor copyright deeds based on their business models. They need to spend to maintain a constant vigil on potential breaches. Companies that work under CCM spend humongous amounts (of money, but effectively of energy) on intellectual property rights (IPR) and vigilance related expenditure. If the desired action is of restriction and protection, the care to be taken is expensive. On the other hand, if the desire is to grant freedom and distribute the rights, such expensive managerial and legal infrastructure becomes redundant. Copyleft abuse is minimal and can be safely ignored.

This line of thinking is the beginning of a new era of publishing, moving beyond books, into software, where it has impacted a much more powerful and wealth-creating environment. We will discuss the software specific issues in the next section, of how it helped the creation and sustenance of DCM of free and open source software (FOSS).

During the last three decades, copyleft was used and hardly abused, by thousands of authors worldwide. This has been the single "weapon" used against the CCM of proprietary software companies. Its success is evident in its use by Wikipedia, Creative Commons, and of course free and open source software (FOSS) development. Before we take further examples, we will examine the ontology and anatomy of controlling media and message in a digital landscape.

How to Control Digital Space

Since this section concerns the central concepts of the digital space, media, code, and message, let me begin by establishing the sense in which they are used. The term "medium" refers to the means and the mode of storing and transmission

of encoded human actions. I will use the term "code" for the encoded human action. The "new medium" refers to the digitising means, and the mode of storing and transmission of encoded human actions. The interpretation of the code (decoding) is unfolding of the encoded human action, which is inherent in the message of the code.

The computing model used for encoding, decoding and transmitting are known to the experts in information and computer technology (ICT). Since this is an area of the new computer science, it seems logical that we should leave it to them to solve the problems arising out of this digital communication space. However, as we will see, we will be in great trouble if we do so. Most of these experts are not working alone. The agencies for whom they work, and the social and economic motivations of their agencies, demonstrate what they do to this space. This section will expose the ontology of operations of this space. By situating these operations in terms of CCM and DCM, we lay bare the social outcomes.

The political operational space of digital society can be best understood if we focus on what happens when we digitise any document, whether text or other kinds of media like audio or video. Digitisation uses a computing model to write (encode) the data in any computer memory, and when we try to retrieve the data, the computer reads (decodes) it for us in a human-readable form. We need not go into the technical details of how this is done, but focus on these two operations for a while.

Since any code is by nature arbitrary, each company can invent its model of digitisation, and provide a computing service to its customers. If the arbitrary computing model they use is not published, decoding it becomes a private, or proprietary, process.

Let us now consider two agencies. The first agency, call it D, produces a document using a computing model where the encoding and decoding model is published. The second agency, call it C, creates a document where the encoding and decoding model is held in private custody.

Spread the documents produced by both these agencies all over the world, perhaps by uploading on the World Wide Web (the common standard for machine accessing and/or publishing human-accessible content). If someone wants to decode (read) the documents produced by D, we need an interpreter, often a discrete software programme. If the software is not made available, one can create one based on the published computing model. The burden of interpretation is not on D. The burden of interpretation of decoding a genetic sequence does not lie on those scientists who cracked the genetic code. Since the method of cracking the genetic code is published, anyone who intends to do this can train oneself. Thus publication distributes the power to decode to the commons.

No one can decode the documents produced by C, unless C creates a software interpreter and publishes the interpreter, either gratis or for a fee. Since the computing model is held private, no other agency can create an interpreter for it. C becomes the center of producing interpreters for privately encoded documents. Remember, the documents themselves are not held privately, it is only the interpreter that is kept under wraps. If you have C documents in your

hand, you will also need a C interpreter, without which the C documents are like Rosetta stones. You have them in your possession, but you have no way to decipher them. Thus for any practical use of C documents, the users depend on a centrally produced (and controlled) interpreter.

One may think that C is protecting innovation by keeping control of the interpreter. However, digital encoding as mentioned earlier is arbitrary. Arbitrary novelty is not an innovation. The computing community realizes this. Therefore, the useful computing models are published as standards. We have worldwide organizations that publish standards like ISO, IEEE, Oasis, and W3C. These standards can be used by other agencies, either for a fee or gratis. The fact that they are available to any agency is good enough for distributing power to the agencies, instead of holding it in one's custody.

The picture becomes a little more complicated when we bring in the required layers of interpretation. In an operating system, there are multiple layers where interpretation happens.

When programmers write instructions, they do so in one of the programming languages. These instructions can be read only by trained programmers or by the specialized compilers or interpreters. They know how to parse them and decode the meaning of the instructions. However, a compiler is not capable of carrying out the instruction, because it is the processor of the computer that carries out the instruction, it merely acts as a barter, by providing an explicit mapping between the programmer and the computer. Since a computer does not understand the human-readable programming language, the compiler decodes our instructions and then re-encodes (re-writes) in a language the computer can decode (understand). Such a rewritten code is called compiled software, which is written exclusively for machines. Therefore it is often called machine language. This code is humanly impossible to decode (though in principle, of course, it is possible). Since there are several kinds of hardware processors made by different vendors, the compilers have to create different versions of software suitable for each processor. Thus, a program made for an x86 processor is not suitable for a PPC processor.

When software vendors distribute their software, they distribute it for a particular machine, and the code that is distributed is not the code the programmer made for the company, but a compiled version. Added to this complexity is that the program cannot be directly passed onto the processor without an operating system (OS). An operating system is another mediator that helps convey the instructions from the program to the hardware, and vice versa. It is therefore also necessary to keep in mind the OS for which the programs are compiled, apart from which processor they were compiled. Sometimes, it is possible to use the same bytecode on all the operating systems, provided that there exists a separately compiled interoperable interpreter for each processor and OS. Languages like Java and Python, for example, work this way. This makes the code inter-operable.

Now that we briefly looked at how our instructions before reaching a processor get re-written into a series of 'languages,' we can now see how to convert software into proprietary software. If anyone wants to take control of the soft-

ware (code) that is made, a proprietary software company can do several things. One way is to provide the source code of a program to everybody, but restrict the interpreter (say a compiler of that language) to only those who pay a license fee. It is also possible to embed the interpreter in hardware, and whoever has the hardware can make use of the software (e.g., iPod). Another way is to provide neither the source code nor the compiler, but only the result of a compiled software (e.g., Microsoft Office), and restrict the operating system to only those who pay a license fee for it (e.g., any of the Microsoft Operating Systems).

It is possible also to restrict the use at all these multiple stages — the compiler, the operating system, and the access to the compiled code. The possibility of restriction does not end here. It is possible to create special hardware, which may also contain another layer of an interpreter, and lock the hardware to a single user who enters into a license agreement with the manufacturer (e.g., controlling at the bios layer which software could run on the board). The early Indian language typing solutions use this model, ironically including those produced by government agencies, ignoring or bypassing a 'policy decision' to honour the principles of free and open publishing. It is possible to invent more and more such stages of control by C agencies.

However, in all these stages, mostly the interpreter and sometimes the code is kept under control. Since code per se is of no value, the process that makes it valuable is the interpreter, which decodes the meaning contained in it. By making the interpreter a scarce commodity, it is possible to enhance its value. Since even interpreters are codified instructions to the computer, one copy of an interpreter can be copied to make several copies. That is why a proprietary software vendor searches for technical innovations that prohibit copying, or to find other ways of prohibiting copying. One standard method is to de-couple the interpretation process into two or more layers and embed a part of that into hardware (e.g., Apple). This way an interpreter can be made a scarce commodity, making it available to those who can afford to pay a license for it or buy the hardware and software together.

Another significant way of controlling is to write user's creations (such as digitized text, audio, video, etc.) in a specific language that can be interpreted only by the system that created it. Restricting the user to use only one kind of application all through their life is the most popular way of enhancing the value of software. Microsoft's 'doc' format is an excellent example of this.

The companies that indulge in this kind of practice provide a justification, which is to collect a fee for the interpretation, claiming the ownership of the interpreter. The money users pay therefore called as a license fee, and not the price of the software. This tactic proclaims that it is a service oriented business. Mind you, and they alert us in the fine print that a very few of us read: the customer is not the owner of the software, and is merely granted a license to use it for a purpose.

Currently, the human effort on the operations involving code is slowly getting taken care by the artificial agents (computers). Owners of the machines are demanding more, instead of asking lesser compensation. Demanding more compensation is not justified, since the precious human effort has come down

substantially. As a result, in place of the substantial decrease in the compensation of human time, they sought an increased fee for artificial agents. Since the new software technology got a fashionable image that it will be creating more value, people began paying fees as per the demands of the software producers. This, to my understanding, increased the wealth of the software 'industry' severalfold, virtually minting digital money. In economics, this production of money without a commensurate increase in underlying value is called inflation.

On the other hand, the hardware is also getting embedded with an increasing amount of software. Increasingly, even hardware is entering into a licensing regime. Software and hardware industry are together creating more and more artificial agents. The main problem with this model is, society began to pay for the services of the artificial agents. The manufacturers of these agents are pocketing the money in the name of the service time of these gadgets. In this new economy, it is not the goods and service time of human agents that are on sale, but the service time of the gadgets. Do the manufacturers of the devices deserve to extract the compensation? Yes. However, only if they do not insert additional locks, that is if they do not prevent free dissemination of cultural resources.

Technical innovation should go towards a way of finding out how to preserve cultural resources for a long time, rather than decrease their lifetime. What is the problem with this business model? I think the problem lies in charging for the service of an artificial and copyable agent (interpreter). As long as the interpreters were human beings, we sought to buy their time when we needed an expert. Currently, most human expertise is getting re-written as programmed instructions, and are interpreted by the artificial agents. The scarcity of artificial agents is controlled artificially by the C agencies, so that their demand increases.

It is important to realize that there are two kinds of artificial agents: the hardware and the software. The hardware is a substantial thing, fabricated generically to carry out programmed instructions. It is not possible to make copies of hardware without spending considerable matter and energy. However, a software set of instructions, on the other hand, is copyable with minimal effort and high fidelity. Writing programs is a creative act, just as writing a formula to calculate in mathematics. The compensation should go to the author, and not to the agency that copies the program. Often programs written by several authors is collected and compiled to produce a re-written form of the program, which the author too has lost the freedom to interpret. The author of the program lost this right. The only way to regain it is to keep the entire compilation process accessible. By becoming a custodian of the latter stages of converting the program into machine code, and its interpreter, proprietary software industry invented a technical method of taking away the right to know. This is not required for making the technology work, but only needed to promote business interests. As was described in the previous section, the code is eminently and naturally copyable. Copyability is code's essence. When people indulge in such a natural process, the C agencies called them 'pirates'. However, this aspect of the work of C agencies is inherently inflationary, which, in societal terms, is arbitrary and exploitative.

To understand that this model of exploitation does not happen only in digital society, let us take a look at Indian history. There was a time when the traditional wisdom in India, often called Vedas and Upanishads, was available only as spoken (verbal) code, and was part of common knowledge (folklore). Later, this wisdom was rewritten (re-encoded) in an artificial language called Sanskrit. Sanskrit is artificial because of the generative structure of its grammar. There is a sense in which all natural languages are artificial (artificial = made by humans), but Sanskrit is not natural, in the sense that it is a rule-based construction, using the same model as our programming languages. It is not the vocabulary that makes it Sanskrit, but the manner in which it combines the vocabulary to generate more meaning.

After re-writing the traditional wisdom in Sanskrit, it became accessible only to those who spoke or wrote this language. It was the elite section of the society – the Brahmins – who had this access. They were the 'compilers' of Sanskrit, so to speak! There was a time when the right to learn Sanskrit was prohibited, by promulgating a rule that only the royal caste (kshatriyas) and scholarly caste (Brahman) could decipher what was in there. Even kshatriyas were prohibited from accessing some portions. This restriction to knowledge was accomplished by creating a private language. This is very similar to the way proprietary companies are making private languages to prevent knowledge from flowing freely. Brahmins called a shudra (person belonging to the lower caste) a *papi* (a sinner), just as proprietary world called those who copy software 'pirates'. Since knowledge is taken to be a beneficial aspect of the human condition, and necessary for survival, in gaming survival it has became part of a political process. The powerful people always harbored the 'compilers,' buttressed them, so that the rulers could remain rulers forever. In computer software, this is the same game. Old wine in new bottle! A corporatised multinational version of brahminism!

The question is, are these mechanisms to take ownership of the code or knowledge ethical? Are they even necessary for doing ethical business? Arguably not. In the last ten years or so, a large number of D companies worldwide began distributing software and provided services around it, e.g., Redhat, Debian, Mandrake, Ubuntu, etc. Instead of selling software per se, or claiming ownership of their creations, they sell the services of experts. Indeed, they also harbour a large number of developers in their business houses, to create more software and enhance the efficiency and value, without claiming ownership of what they produce. This proves that ownership of tacit property is not essential for business. The rise in their service business is evidence to its success.

The objection is not to making money, but to the means of control. This is not using technology to enable society to access knowledge freely, it is using technology to curb free access to knowledge. Their inventions are counter to preserving human values.

The story does not end here. There are other aspects to it.

To understand how criminal these intentions are, let us do an activity. Some of you readers may have been using computers for the last 10 to 20 years. Collect all the documents you created during this time. Attempt to open those

files today, using your modern OS. Take my word, several of you will find that most of the older documents are unusable. Either the document cannot be opened, because the software that you have on your modern OS does not know how to decode the older documents, or after decoding you see that there is some significant loss of data.

Now the question is, why did this happen? This because the underlying encoding technology was modified. If our work is inscribed in digital media like this, how can using computers for storing data be justified? Printed books, old audio tapes, etc can still be used, because they are not encoded in a secret language. We have not lost the ability to decode them. However, our modern operating system lost the ability to decode some of the older computer documents that the same system once produced. Who is responsible for this loss? If the agency which lost this is a public body, like a government, who can be held responsible for this loss of data? Often enough, people have been using, all along, software made by the same company. Yet, this loss took place. Why?

The answer is not technical. True, advancement in technology requires that there be some changes. However, what is the justification for changing the encoding of data, without converting them to a newer form? The computing industry has norms to follow, such as encoding standards, e.g., ASCII, Unicode, XML, HTML, etc. Modern computers did not lose the ability to decode the older encoding standards. The loss took place because the software was proprietary, and the encoding of the data was also proprietary.

If using computers for our work means such loss of data, doesn't this become a sufficient reason to stop using computers? Even our inscriptions in the ancient caves still exist, although deciphering them is often a challenge. This indicates that preserving code is not enough, and we must also preserve how to decode. The only way out of this problem is to make sure that we save our work in a standard format, and record in the museums the process of decoding. Currently, our museums store only the code, often forgetting their meaning. Most users of computers may not be able to understand these subtleties of code dynamics. In such a situation, it is the responsibility of educational institutions, media, and the companies themselves to advice the users to follow the best practices. There must be governing policies to preserve cultural records, and not to preserve knowledge in a proprietary format. Otherwise, they need to, at the very least, include a prominent warning, that there could be a loss of data, if all of it is not regularly, even frequently, recreated and stored anew.

If we collect all the digital documents which cannot be usable in today's operating systems from all the users from merely the last fifteen years, we will realize that this is not an ordinary loss. It is nothing short of wiping away history, since it is the documents that contain what we did in the past. If the documentation cannot persist, we cannot preserve history. It is like walking on quicksand, where we find it difficult to trace the steps we took. Who is going to pay for this loss?

Having seen how the C agencies are controlling the digital space, let us see how the D agencies do it, by employing the copyleft model in the next section.

Software Freedom Movement

As we saw in the previous section, several layers of possible exploitation of digital versions of our cultural practices are at stake due to holding interpreters (decoders) as private property. To prevent this Richard Stallman invented the idea of copy*left* which is an inherent part of the very definition of *free software*.

We realize from the discussion above that software is nothing but a language, though it is invented artificially by a small set of programmers, unlike a natural language. In the computers that we use, several layers of language and different kinds of them are supported, each with their syntax and semantics. However, once we accept that software is a language, we treat software as a *creative expression.* A legal way of protecting a creative expression of ideas is by applying copyright. Therefore the focus of the discussion turns to who holds the copyright and what are the terms and conditions of the copyright. Richard Stallman did not challenge the legal instrument of copyright, instead asked us to modify the terms and conditions. The inventiveness of his proposal consists in ensuring that we do not curtail the freedom to interpret at any stage of creative expression. *Free software*, thus, is software with liberty to encode and decode.

The first thing we need to know about the term "free software" is that its meaning does not arise from the combination of the terms "free" and "software." The meaning of this term arises from the definition, and not from the terms it contains. The term "free software" is defined by Stallman as that software which gives the user the freedom (1) to use it for any purpose, (2) to know how it works, (3) to improve it by modifying, and (4) to share or propagate or distribute the modified code to others, provided all these freedoms apply recursively to all distributed copies. This is the essence of *General Public License (GPL).*

Any software that meets these four criteria can be called free software. We must notice that there is no mention of the price of software in its definition. This means that there exists a possibility to pay or charge for software. Since free software is intended to give the users the freedoms mentioned above, it is better called "freedom software." In Indian languages there are more options: we may call it "swatantra software" (a preferred term in southern and western India), or call it "ajaadi" software (a preferred term in north-eastern India), or else call it "mukta software" (a preferred term in northern India). The last option is nice since we can create a near pun with the word to say: we are talking about mukta, and not mufta (gratis) software. Let us, therefore, bear in mind that free software is not about price, but it is about freedom.

A necessary implication of software freedom is an invitation to shape the technology by anyone who respects the freedom to encode and decode. As a result, the GNU project, founded by Stallman in 1984, unfolded into a full operating system used in almost every computer in some form or the other. Several geeks, often called hackers, collaboratively created several programs and published the code online. Other users, who may not have been that proficient with coding, contributed documentation and user manuals. Others translated them. They made programs 'speak' all the languages of the world. People made several modifications and customisation of the programs, which expanded their

diversity, and eventually, natural selection worked on this diversity of programs, to create the best quality programs for any given purpose. Today, GNU/Linux operating system is leading by holding about 80% of the Internet servers, from small systems to supercomputers. It is also spreading fast in mobile and desktop computers.

An exciting development was that the geeks also created software platforms that facilitated collaboration, by publishing information about who contributed what, and when. Transparent auditing of not only the programs, but their process of development, was made possible. This demonstrated the possibility of an alternate form of bureaucracy, or if you like, the elimination of bureaucracy.

This prospect became part of another widely acclaimed project, Wikipedia. Wikipedia used precisely the same model of collaborative development of articles on every topic and in every language of the world. Wikipedia uses free software to produce free knowledge by employing a transparent bureaucracy. Nothing is hidden, the entire process of how each article is written is also part of it. For the first time in the human history, the history of each creative expression is also encoded.

Another impact of this movement is open access journals. Many of the traditional research journals bargained to take the ownership of copyright from the scientists and restricted the journals to only subscribers. Open access journals used a derivative of the copyleft model, popularly called *Creative Commons* that modified the terms of the copyright, similar to GPL. Though open access journals are a welcome development, they are far behind in documenting the process (history) of science, since they provide open access only to the generated product and not the process. They have much to learn from Wikipedia and free software projects.

Implications to Distributed Justice

Several areas of our lives are affected by digital technology. Most widely used devices are a result of digital communication technology, in the form of mobiles. Most mobiles are 'infested' with proprietary software, that extracts information about users without our knowledge. They are Trojan horses living in our homes and pockets. Whether it is set-top boxes or mobile phones, all of them have encroached upon our private as well as public lives. Only recently several countries woke up, and have begun to regulate this space. India is also discussing a draft law for the protection of private data, as we write this article. However, by confining this protective discussion to 'data', meaning in fact digitally stored data, this is a thinly disguised assault on personal freedoms enshrined in the umbrella term privacy. Privacy was, as it happens, reaffirmed as an inherent Constitutional right following the imposition of a digital numbering scheme, branded 'Aadhaar', that, in effect, compromised the societal understanding of identity in India, and which has been subsequently significantly curtailed.

Though we are a professed democracy, the recent use of digital technology by the Governments does not indicate they are interested in distributing power

to people. The Aadhaar project, a centralized repository of providing UID to all the residents of India, goes contrary to the idea of freedom. The identity of a human being is socially constructed. Governments are denying the self-regulated identity by increasingly moving towards centrally granted identities using biometry. Linking this to various services amounts to a denial of service. This is a blatant abuse of ICT by the state to create centralized, instead of preserving the existing decentralized and socially constructed identity. All the problems associated with centralised systems discussed above make this system vulnerable and expensive. Free software developers have demonstrated how signing (endorsing) each other's electronic signatures produces greater distributed trust than a centralized trust. Centralized trust can become corrupt very quickly due to single point control. The Aadhaar story so far demonstrated us beyond doubt that greater the linkage of UID, the greater will be the leakage of personal information. It is unfortunate that the Supreme Court of India does not see this, because a majority of the judges assigned to the Bench adjudicating it, except for the dissenting Justice Chandrachud, have also been caught napping.

Digital technology provides several ways of enhancing transparency which could minimize corruption in all walks of life. Governments are still using several layers of proprietary software for running the daily functions. Some of the geeks who exposed the vulnerabilities of proprietary software running on Govt platforms, voting machines or Aadhaar data leakages were arrested, instead of giving them incentives for protecting the system. This attitude of the government and the policy makers indicates that the so-called democratic governments function like centralised powers. Wikileaks demonstrated how such governments could be found behaving corruptly. They mostly abuse power.

Digitised knowledge is eminently copyable, making it no longer a scarce commodity. Similarly, a deregulated spectrum is necessary for the last mile connectivity of Internet. However, government sits on this abundant medium, artificially increasing its value by treating it as a scarce commodity. This is the most brutal act of preventing people control of their own media, the remote (or distance) counterpart to speech. Controlling resources that are not abundant, or are not recyclable, such as oil, makes sense. This case of centralized control of spectrum makes governmental behaviour similar to a mafia. In the name of security, the military and State are squatting on this most abundant natural resource. No government anywhere in the world grants a license to the commons, except for the narrow WiFi range of the spectrum. This license model is no different from a mafia holding a *basti*, collecting *hafta* in the name of protecting the small-time vendors doing business (the protection racket), or the British colonial government taxing Indians for producing salt from the natural seawater. In the name of security, spectrum modulation rights are held by the mafia (Government and large Internet Service Providers). Even within the deregulated spectrum of WiFi, people are not allowed to run ad-hoc mesh networks, because Government cannot control them. Do we have to do spectrum *satyagraha* to gain freedom from licensing spectrum?

A new and very dangerous game by software companies is to provide gratis proprietary software in the name of social networking and communication appli-

cations like Google Search. Gmail, Facebook or Whatsapp, which are used by billions of people around the world. These are Trojan horses. Even if the users levy a fee to the company, the business they do is still profitable. They earn revenues by selling our profiles, without paying us for giving our information. All policymakers across the globe are caught napping in this case. Stallman warned about this danger several decades ago.

It is not only the policymakers that were caught napping, the syllabus makers of schools, colleges and universities also were, by introducing proprietary brand names in the syllabus as well as in the examinations. Instead of testing the skills, they test whether the students have rote-learned a specific brand of an application. It is like forcing students to use only pencils of a particular make in an examination. Certificates issued by branded companies like Cisco, IBM, Microsoft, Adobe, etc. are gaining higher value than certificates that display proficiency of generic skills. This trend is not towards freedom of expression.

The danger of using proprietary software in any area of life is treacherous. It is profitable only for big corporations and power hungry governments. If we believe that the direction of democracy is towards distributing power and not accumulating power in the name of governance, we should criminalise their use everywhere.

Computer scientists have demonstrated how economical and efficient will be the transfer of documents when published and transmitted in P2P networks (torrents). Since this is not in the interest of ISPs, government and ISPs criminalize these networks in the name of 'piracy'. Just as spectrum is controlled in the name of security, the P2P network infrastructure is denied to people in the name of piracy. This is a clear illustration of whose interests the government supports, over the rights of free citizens. Complete denial of democracy is illustrated. In Myanmar, free (unbilled) access to Facebook on mobile phones has led to the spread of hate speech, resulting in the massacre and expulsion of large numbers of Rohingyas, a community in the northwest of the country.

Inter-governmental bodies meet periodically to regulate the space through treaties, expanding and protecting the mafia. Digital Millennium Copyright Act (DMCA) of USA, which criminalizes production and dissemination of technology to circumvent access to copyrighted works, but at the same time does not criminalize the production and dissemination of technology for Digital Rights Management is a clear indication of which side Govts take. It is a myth that Govts are for, by and of the people. Isn't Cambridge University Press's republication Gandhi's Hind Swaraz, with an added editorial, a circumvention? Go to a nearby book store and look for William Shakespeare's books, all of them in public domain due to expiration of copyright, are available as copyrighted works because they have been either abridged or another expert inserted a commentary or a preface. Isn't this circumvention? Does DMCA apply on this technology? The Internet was born as a DCM, but private corporations are seeking Government's sanction to convert the Internet into a CCM. The Internet is governed initially by self-regulating protocols. Frequent demands by various Governments to block social networks, microbloging sites, WikiLeaks, stop the Internet in politically dissenting areas, etc is an indication of how it would work

if they regulate the Internet. This is a clear illustration of where governments are heading. There are countries that prohibit any political dissent, whether it is in a town square or on Internet 'square'. The governments do not control the free software world, or the Wikipedia world, but they do produce what people need. This demonstrates that a transparent protocol based administration is possible without centralised legislation. It is not an accident that only proprietary operating systems harbor computer viruses. Centralised governance, by its very nature, is bound to become corrupt. Power corrupts people, when not negotiated.

Information and Communication Technology (ICT) is a genuinely empowering medium and can hasten the process of distribution of power reaching the goals of a democratic polity. Central control of anything, including ICT is expensive, unscientific, and also inefficient.

The story of the free software movement gives us hope that the means of distributed power is not impossible. Commons can make, share and manage small to gigantic scale projects without centralised control. However, this may not be acceptable to power-hungry political parties. They ask for strong and stable government in their election manifestos. What we need are weak, dynamic, transparent, sustainable social systems that are grounded in ethical protocols, rather than strong legislation and executive. A weak government is a blessing for democracy, not a problem to fix.

The Free Software Movement is a political and cultural intervention in an apparent disguise of technical practices. The movement does not become a Trojan horse, because it started with an explicit manifesto. This is already evident in the way how it impacted distributed development methods, transparency, collaboration, social networking, the emergence of creative commons, p2p governance, etc. If these episodes are considered primarily geeky by the policymakers, they will be caught napping once again. It is time to study these highly successful methods of self-governance, the true meaning of democracy. Let us conclude at the end that there are multiple lessons to be learned from these episodes, and each of them has policy implications. They deserve to be attended by the policymakers in the Niti Ayog.

To Conclude

Socio-political problems cannot be fixed by technology, however technology can facilitate and expedite the distribution of justice or disruption of social fabric. Copyleft cannot fix freedom of expression, but it will make it easy and accessible. Copyleft cannot fix abuse of media, such as distribution of indecent content or trolls. However, technology can help detect them through transparent collaborative models, and fix it faster than a centralised model where we appeal for justice through a hierarchical judicial system. Copyleft is a paradigm case of how justice can be made *inherent* as a protocol. Technology cannot grant RTI (right to information), but it can make RTI redundant by making information always accessible. Who will need to file an RTI on Wikipedia? Technology

cannot inform us that freedom to whisper is a political necessity and so must be protected. However, technology can help us create means of efficient whispering. The idea that personal data should be protected by the state does not come from technology, but mismanagement of technology can make the personal data leak without notice is a technical lesson.

Technology that we use not only determines but also expands the action field of human beings, individually or together as a group. And so, it can transform the existing fabric of society, and often this transformation could be disruptive. Whether the action is ethical or not, is part of continuing social mediation and negotiation. Though we cannot fix this through technology, if the technology we use is proprietary, we cannot know what is being done to us in our own action space. That is why, we should never allow opaque technology to enter our lives. Public audit of all technology used in public space for a public goal must be made as a mandatory protocol by the state. Computer Hardware without open drivers should be treated as Trojan horses.

Technology grants an extended action space, which become power to those who have access to technology. In turn, this power could be used by forcing a person or even a country to become subservient or fight. Negotiating this space is the story of human history!

Human actions facilitated by technology can create resources that have exchange value. Multiple forms of currencies/coins are emerging, several of them digital, e.g. Bitcoin. The technology to mint or participate in an exchange of those coins does not seem to be accessible to commons. All of our enthusiasm to distribute justice could collapse, when a new form of currency gets into our life, while we are caught napping, which could take over all our negotiation space. On the one hand, we may feel good that corrupt governments do not have a hold of this space, but on the other hand neither does the commons. This does not seem to be a game for equity. These are also serious issues that we need to negotiate in the space of media politics.

The bottomline is: the human action space, without technology, is unimaginable. We are what we are because of our ability to create and use technology. Politics without technology, digital or not, is non-negotiable. What is negotiable is which actions are justifiable.

References

Achal, P (2012, January). Would Gandhi have been a Wikipedian? - Indian Express. Retrieved October 27, 2012, from http://www.indianexpress.com/news/would-gandhi-have-been-a-wikipedian-/900506/

Cohen, D. J., & Rosenzweig, R. (2006). Digital history. University of Pennsylvania Press.

Pollard, A. W. (1922). Some Notes on the History of Copyright in England, 1662–1774. The Library, 4(2), 97.

Stallman, R. (1985). The GNU Manifesto, 1985. http://www. gnu.org/gnu/manifesto.html.

Stallman, R. (2002). Free Software, Free Society: Selected Essays of Richard M. Stallman, Free Software Foundation.